BABEŞ BOLYAI UNIVERSITY CLUJ NAPOCA

FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

SPECIALIZATION SOFTWARE ENGINEERING

DISSERTATION THESIS

Real-time human pose estimation and tracking for rehabilitation therapy

Supervisor
Lect. Dr. Ioan Lazar

Author
Razvan Timis

2019

Table of Contents

List of Figures

Abstract

The goal of this paper is to obtain reports on patient progress based on intelligent pose estimation algorithms motivate the patient to continue medical recovery.

Most patients require long-term medical recovery, which means an average of 2 or 3 years, give up during the recovery period because they do not see results even if they exist, they are not obvious to the naked eye. More worrying is that up to 70% of patients give up physiotherapy because they can not see immediate results.

Our solution consists of making two prototypes demonstrating the ability of these algorithms to integrate into a solution to help the end user, we will focus on the performance of these algorithms.

The main problem is low performance because it requires a large amount of computing. The objective is to run algorithms at a performance of 30 frames per second on the end-user device. Next, we studied two important architectures in pose estimation.

MobileNet is optimized for mobile, reducing the number of calculations through special architecture [11].

Convolutional pose machines that specializes in pose estimation, having a composite structure of several classifiers [24].

In this regard, we have chosen to use two separate applications of these architectures, observing the performance and accuracy of the estimates.

The mobile application for tracking patient progress in real time using the camera phone. He will do each exercise in front of the camera and receive reports related to his performance on days. The performances are the distances traveled during the exercise by its members.

The web application will calculate a range of motion to help the patient perform the correct exercises, so we will count the correct number of repetitions of an exercise based on the angles that help the patient during the exercise.

Following the implementation of the two variants, we discovered that the network architecture presented in the paper with name Convolutional pose machines [24] is more powerful and accurate than the web implementation using the MobileNet architecture, which is not particularly optimized for the pose estimation problem.

This work is the result of my own activity. I have neither given nor received unauthorized assistance on this work.

BABEŞ-BOLYAI UNIVERISTY
CLUJ-NAPOCA, JUNE 2019
Timis Razvan Vasile

Chapter 1

Introduction

More and more people need medical recovery from all age groups. There are many cases where there is a medical problem that can be resolved by medical recovery.

For example, in the case of people who suffer accidents and remain paralyzed completely or partially, recovery is vital in maintaining and improving the person's condition. Other problems that occur frequently at any age are fractures that require increased attention and a recovery program in most cases.

In the case of older people, there are many problems that arise and require medical recovery, one of the problems is osteoporosis, around 10 million Americans have this problem and another 34 million have problems with bone mass, which is a high risk of a problem.[13] And in the case of children there is a significant number that is born with different deficits of movement.

Many people quit medical recovery due to lack of progress or high costs.

For example, America spends more than $18 billion annually on account of injuries to older people (Burns, Stevens, Lee, 2016; Stevens, Corso, Finkelstein, Miller, 2006).

Next we will talk about what is Physical Therapy and what are the main goals in a medical recovery program. How to track the progress of several sessions and in the end we will talk about what it means to detect a person's posture.

1.1 Physical Therapy

Ever since antiquity (ancient China, ancient India, ancient Greece, ancient Egypt, etc.), physical exercises have been practiced to maintain a good physical form but also to treat diseases such as muscular pain, gout, obesity, etc.

The great Greek antiquity doctor, Hippocrates, in the first hospital built on the island of Kos, carefully studied the physiological effects of gymnastics and massage defining health as a balance between exercise and nutrition. For the first time, the relationship between motion-hypertrophy and immobilization-muscular atrophy is noted. He argues that physical exercise and massage influence favorably the breathing, metabolism, blood circulation and balances the activity of the central nervous system. [7]

Physical therapy is used for somato-functional medical recovery and consists of a set of techniques and methods that center the physical exercise.

Physical therapy includes three forms of kinetoprophylaxis: primary, secondary, and tertiary.

- Primary when using physical exercise, techniques and methods to prevent illness.
- Secondary kinetoprophylaxis has the role of preventing complications of illnesses.
- Tertiary kinetoprophylaxis prevents the appearance of sequelae following illnesses that could cause motor disabilities.

The main goals of physical therapy are relaxation, increased exercise capacity, increased strength, muscle strength and posture correction.

Current technological developments have allowed the use of computer technologies in physiotherapy for different measurements, allowing accurate assessment and treatment personalization.

1.2 Track progress in rehabilitation therapy

In Physiotherapy, tracking Range of Motion (ROM) is a standard approach to measuring progress in patient therapy. Often, ROM is measured subjectively and documentation is inconsistent between clinicians. Physios might come to wrong conclusions if ROM is tracked incorrectly between therapy sessions.

The problem is that up to 70% of patients give up physiotherapy because they can not see immediate results [14].

How do we measure range of motion? For example in Figure 1.1, the elbow of the hand has a range of motion that is expressed in degrees So we can say that each specific joint has a normal range of motion that is expressed in degrees.

Often there is a problem due to connective tissue, which is involved in the process of repairing the body after a trauma or surgery and causing the limitation of normal joint motion.

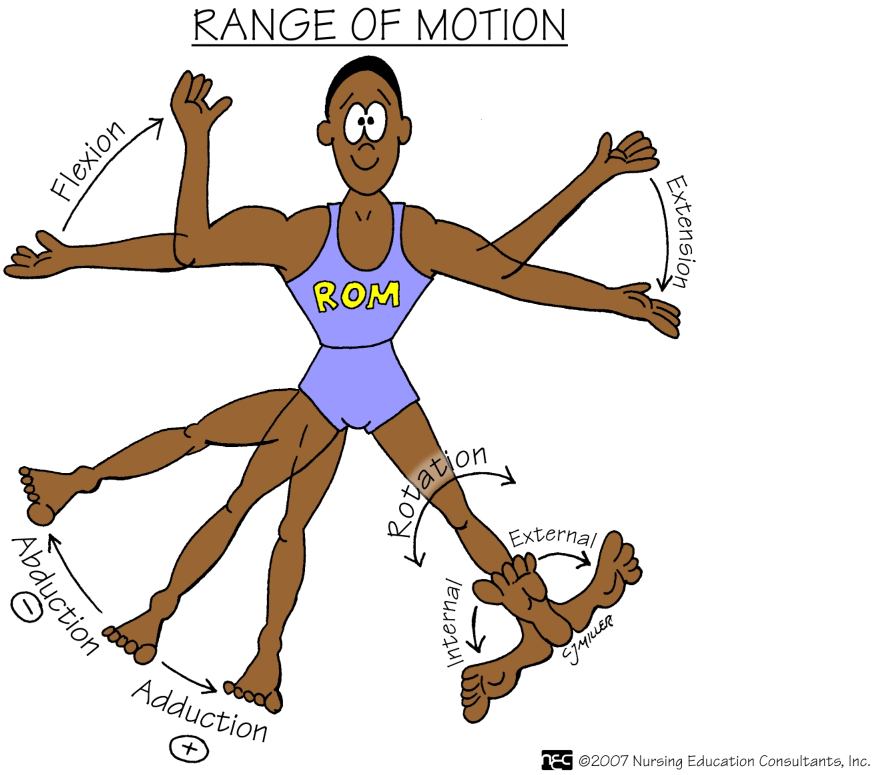


Figure 1.1: Range of Motion

That's why we want to make a application which makes use of a camera to objectively calculate ROM in real-time and automatically produce a report that tracks progress over the course of several therapy sessions.

Another approach we use to measure the progress of patient movement is to measure the distance of the patient's hands and feet. For example, if the patient performs an exercise session after the session is completed we will be able to see a report with the right and left hands, as well as the legs.

In this paper we will try to implement two prototypes of applications, a web application and a mobile application, in which we will address the two approaches to track progress described above.

1.3 What is pose estimation?

The task of determining the position of an object in an image or even a real-time video, as is the case in the present work, is called pose estimation. Pose refers to the object's position and orientation in the coordinate system.

In the computer vision industry, for the first time, pose estimation algorithms have been developed that have been used especially in robotics to schedule a robot to do certain tasks such as moving certain objects to a warehouse. [22]

The pose estimation problem can be solved in three way:

- Analytic or geometric method consists of identifying a set of control points on the object, usually obtained on the basis of features such as edges
- Genetic algorithm consists in using positions as genetic representation and calculating errors as the value between the projection of the object control points and image based on fitness function.
- Learning-based method uses a system based on artificial intelligence, usually convolutional neural network, in the next chapter we will describe how it works.

In this paper we choose the third method based on artificial intelligence, considered the most appropriate.

More than that, we try to run these algorithms on the browser and on mobile devices, in Chapter 4 we will explain in detail these details.

Below we will present more details about intelligent algorithms that have learned to detect the position of body parts

Chapter 2

Theoretical Background

In this chapter we will present a brief introduction to neural networks and then describe the underlying concepts of the Convolutional Neural Network (CNN). We will focus on presenting the main concepts that will be used in the development of pose estimation algorithms.

2.1 Neural network

In the Neural Networks chapter, we will talk in the first part about a brief history, presenting the three periods through which artificial intelligence has passed. Then we will address the biological part that is found in neural networks. In the next section we will discuss what is a neural network , how many types of such networks are there? What is a neuron? But an activation function? Finally, we will discuss learning such a neural network with its specific algorithms.

2.1.1 History

Artificial intelligence takes us to think of some SF films, but it still has a long way to go, for now these intelligent algorithms can only get to the level of intelligence of an insect, they work better in certain exact tasks as imagine detectie and not in general like a brain.

But let's not forget that this domain has been built around the dream to overcome human intelligence. The essential question is whether such a system can be implemented on a computer?

The domain of psychology was the first to have had the artificial intelligence applicability, the most famous Turing test that appeared in 1950. It involves a conversation of a person with a computer and another person and he has to guess who is the person. [21]

Three great periods are in the history of artificial intelligence. In the first period, only after the Second World War. The first programs that implement various smart algorithms to solve puzzles. [21]

An important algorithm in this field is Samuels' game, it was quite simple to implement. They save certain winning positions throughout the game.[21] The first period kept until 1965, but no algorithm has led to major changes in people's lives. [21]

In the second period, it focuses on the processing of natural language so many applications are launched that implement concepts from the processing of natural language.

One of the famous programs of those times was called ELIZA. This program learns to copy the conversations of a psychologist with his patients. How did ELIZA work? ELIZA had a knowledge base in English and a field of psychology made up of a set of rules, so ELIZA worked on the "fit" principle. For example, when it found the word "father," it said, "Tell me about your family" [21]

Also during this period, rule-based expert systems appeared. One such system was called MYCIN, which was designed to diagnose infectious diseases of the blood and recommend medical treatment. It is based on the rules made by specialists. [21]

From 1975 to the present, there is the third period of artificial intelligence. This domain is becoming more stable and the industry adopting these intelligent systems.

Warren McCulloch and Walter Pitts have been the first research that has made the research. In 1940, they highlighted the first digital model of a neuron that discovers the computational capacity, also providing a mathematical abstraction of this concept. Thus the synapses that a neuron makes through dendrites become inputs, the body of a neuron becomes an activation function, the axon has become the output and the BAIS notion has been introduced for mathematical restlessness features. [4]

In 1969, authors Marvin Minsky and Seymor Papert published the book "Perceptrons," which highlighted the limitations that exist for one-level neural networks. After this publication many people who doing research was quit. [4]

Neural networks have become one of the most used in our days. Big Cloud companies offer an API through which any developer can get his own training in the easiest way.

2.1.2 Biological

The great mystery of this universe is the way people think, knowing for several thousand years that powerful head shots can generate loss of consciousness or even death. But more than that, we know our brain is different from animals. In about 335 BC, Aristotle wrote, "Of all animals, man has the greatest brain." [18]

The nervous system is made up of neurons. Each neuron can be represented as a unit. Between neurons there are synapses that can be of two axo-somatic or axo-dendritic types. The body of the neuron has two types of extensions [10]:

- Dentrites are relatively short, and branched near the cell
- The axon is longer and thicker in the propagation of the electrical impulse.

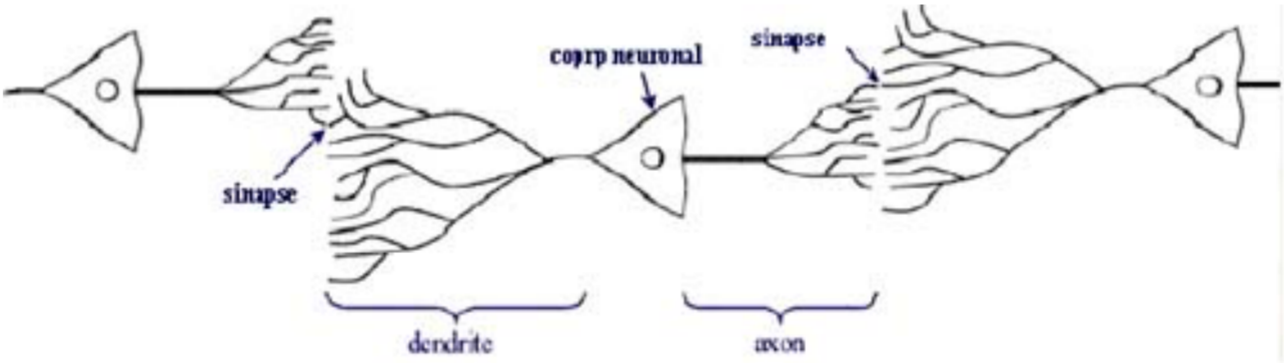


Figure 2.1: Network of biological neurons

While a reader uses a network of 10^{11} biological neurons and 10^4 connections between neurons, the processing time of 10^{-3} ms and 10^{-9} ms for electronic circuits, although biological neurons are more slower than electronic circuits, biological ones can process information much faster than any other circuit. [10]

2.1.3 Overview

A neural network is a copy of the biological model, so it has nodes that are connected by links, these links are actually real numbers that represent the memory of a network. Learning a neural network is accomplished by changing the weight between the nodes.

Any neural network is organized on layers, so there is an input layer and an output layer. The input layer is the one that receives the data from the outside and the output layer provides us with the information we expect. [18]

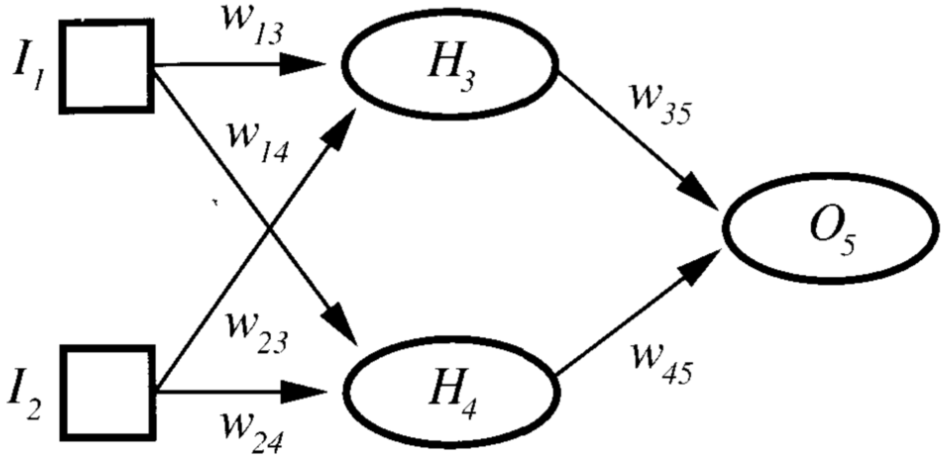


Figure 2.2: Feedforward neural network [18]

In Figure 2.2 shows a neural network of the three-layer feed-forward type, which has links between the neurons in one direction, because the cycles in such a network are missing, the calculation can continue evenly from the input nodes to the exit ones [18].

We can say that everything is summed up to the calculations of a value using input values, weights on each layer, and activation function, most of the times it is difficult to choose the activation function or other parameters that make the performance of a network increase.

If we compare the human brain to such a network, we can see that some regions in the brain are feed-forware, but others do not, with quick back links. Then we can say that the human brain is closer to recurrent networks. [18].

The best-understood neural networks of the recurrent type are thought to be Hopfield networks. Figure 2.3 shows such a network. It can be observed that they use bidirectional connections between neurons, the weights of the neurons being symmetrical. Besides, all layers in the network are both input and output, and the activation function is the sign function. Such a network functions as an associative memory.

Regarding the structure of the network, sometimes the performance of the network is very much dependent on this, because if we choose the wrong structure of a network then its performance will be poor. For example, if we choose a network too small, the mathematical model will not be represen-

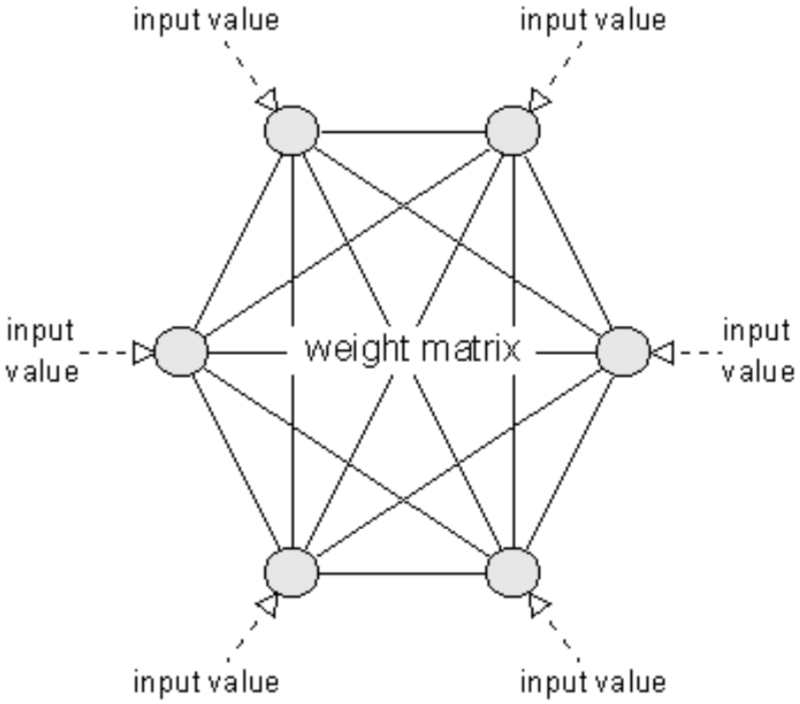


Figure 2.3: Recurrent Networks - Hopfield [18]

tative of the training data, and if we choose too much a network, it will learn the training data well, memorizing them but will not be able to generalize the new entries. [18]

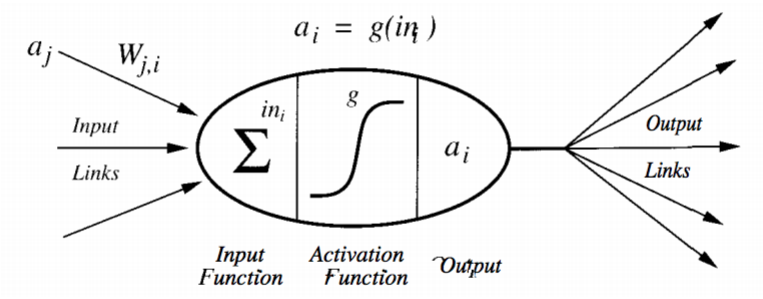


Figure 2.4: An artificial neuron [18]

On top we will describe how a neuron works. Each neuron is connected to other neurons, some of them being input and others output, they make a relatively simple calculation, this computation being divided into two components, one linear and another non-linear. The first is a weighted sum of the input values in the neuron, this is the linear component and the second component is represented by the activation function, g, a_i it converts the weighted sum to the final value, which then serves as an activation value units, most of the times the entire network uses the same activation function [18].

There are different models for the activation function, but only three of them are the most used, step function, sign and sigmoid, shown in Figure 2.5. These activation functions allow the network to model nonlinear relationships between input parameters and output values. The step function returns the values according to a threshold t if the value is greater than the return threshold 1 and otherwise 0. There is also a biological motivation for this function, 1 represents the transmission of the nerve impulse and 0 the transmission failure.

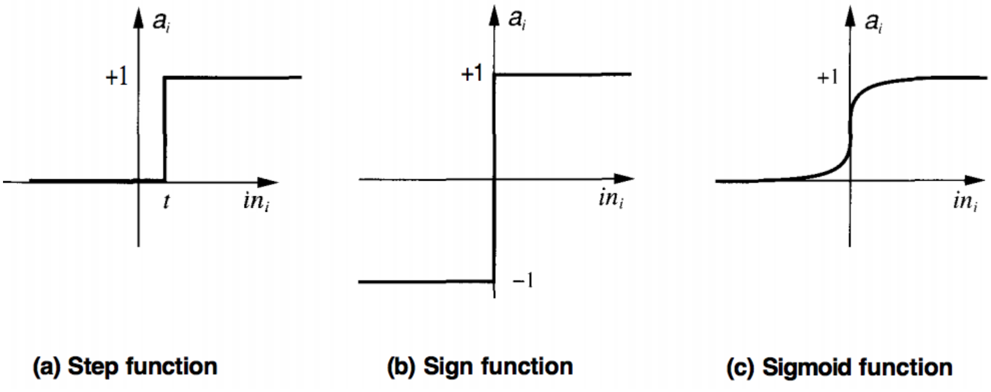


Figure 2.5: Activation functions [18]

Another important issue is networking, their ability to learn by interacting with the environment is the main feature of the neural network. One of the most popular algorithms is backpropagation. We dedicated a special section below for the backpropagation algorithm. The idea behind learning algorithms is based on updating the weights, if there is an error between the output of the network and the value we expect then it is updated weights so as to reduce the error, dividing it equally between weights.

2.1.4 Backpropagation algorithm

It was discovered for the first time by Bryson Si Ho in 1969, since the beginning could not be used due to the large computational requirements. [18]

This algorithm is based on updating the weights, trying to minimize the error between each calculated output value and the expected value. This process is done by propagating back errors. To update the weights, the derived activation function is used.

This weights update process is repeated for all drive data trying to get as little error as possible.

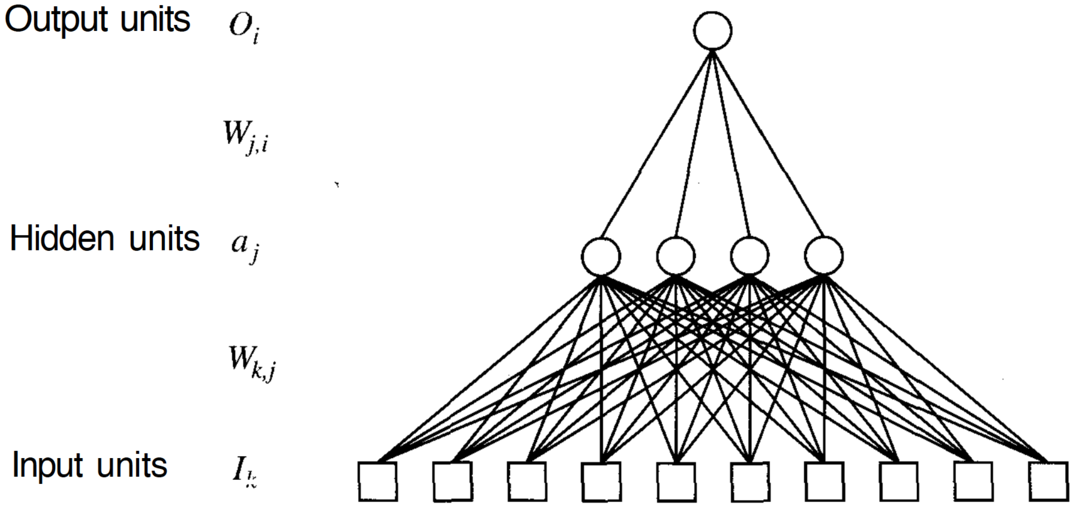


Figure 2.6: Multilayered network [18]

Starting from the idea of updating the weights between the layers. Between the exit entrance and the hidden layer, noted $W_{i,j}$, we will use the formula $W^{'}_{j,i} = W_{j,i} + r \times \Delta^{ouput}_{i} \times O_i$, where r is learning rate and Δ^{ouput}_{i} is calculated by using the formula $\Delta^{ouput}_{i} = Err_i \times g\prime(O_i)$

Now we can calculation of errors by the formula $Err_i = T_i + O_i$, where O_i is output value and T_i is a expected value.

Next we will need to update weights from the next layer using $W^{'}_{k,j} = W_{k,j} + r \times \Delta^{hidden}_{j} \times I_k$ where $\Delta^{hidden}_{j} = r \times \Delta^{output}_{j} \times a_j$.

Backpropagation is used by computers to learn from their mistakes and get better at doing a specific thing, above I described this process. So using this computer can keep guessing and get better and better at guessing like humans do at one particular task.

2.2 Convolutional Neural Network

In this section we discuss the main notions that occur in a neural convolution network that is a kind of neural network but they are capable of processing large amounts of data such as images. The name "convolutional neural network" indicates that the convolution mathematical operation is used. Convolution is a mathematical operation on two functions (f and g) to produce a third function. [17]

In this thesis we will focus on using CNN on images, more precisely on the image classifications.

For example, if we want to classify pictures with dogs and cats, the algorithm will receive an image as input data. The algorithm interprets the input image as a pixel array, like in Figure 2.7.

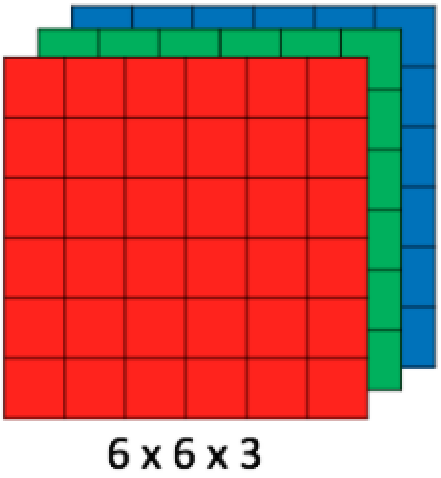


Figure 2.7: Array of RGB Matrix

Next, we will need to define the layers of the network, which can be of several kinds: convolution layers with filters , pooling layers and fully connected layers (FC) and the last time we will apply Softmax function to classify. Each image will pass through these layers to be classified in one of the two categories.

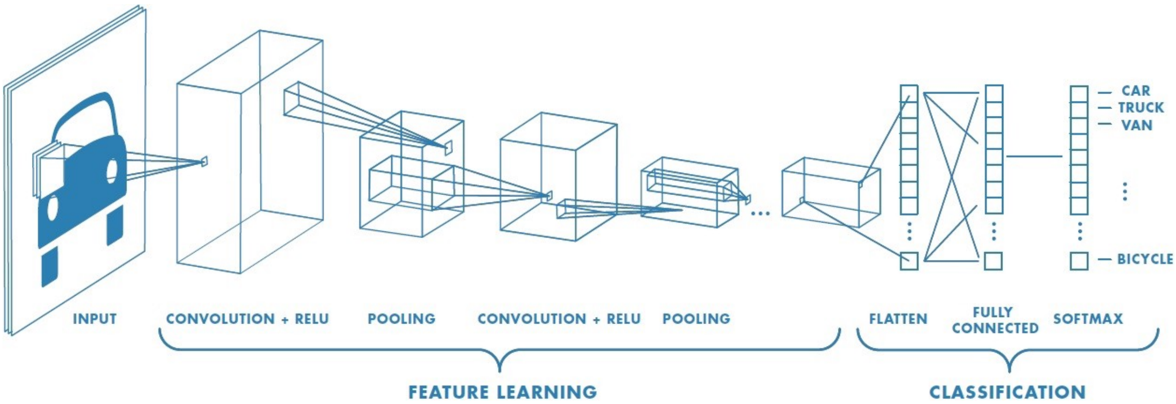


Figure 2.8: Neural network with many convolutional layers [1]

2.2.1 Convolution Layer

Convolution operation take two inputs: imagine matrix, filter matrix. For exemple if we have a imagine 5 x 5 x 3 with a dog and apply convolution operation with filter matrix like in Figure 2.9, we will get the feature map but how? Convolution operation first creates a sliding window of size

$K \times K \times m \times (k-1)$ that goes through height and width dimensions of input imagine and every pixel is multipler with filter matrix and result a feature. [17]

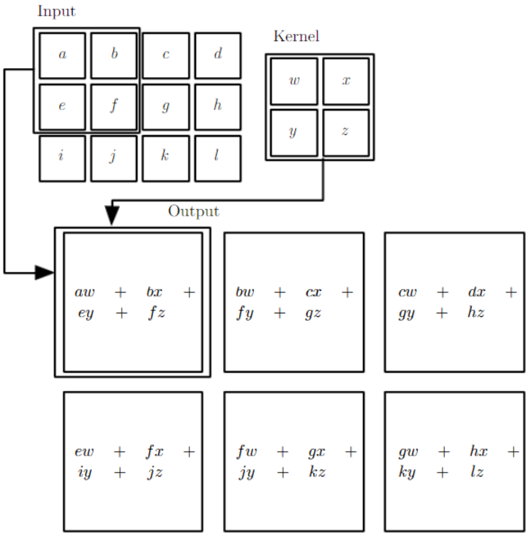


Figure 2.9: An example of 2-D convolution [17]

Depending on the type of filter applied during the convolution operation, it leads to obtaining different feature map. For example, by applying a convolution operation to a image with a dog using the matrix in Figure 2.10 ,operation gives a feature map what describing the edges of the imagine with a dog, Figure 2.11.

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 4 \\ 0 & 1 & 0 \end{bmatrix}$$

Figure 2.10: Edge detection filter

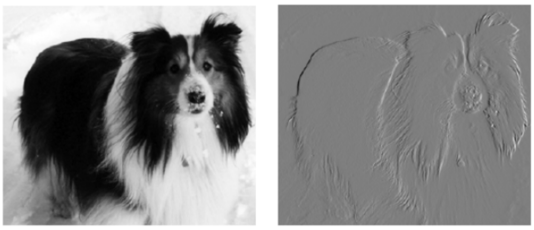


Figure 2.11: Apply convolution on imagine with dog [17]

Convolution of a different image filter can extract different features that help to better learn the neural network, extracting features related to edge detection, blur and sharpening.

For example, to create a convolution layer with tensorflow, use the syntax:

```
conv_1 = tf.nn.conv2d(input, filter, strides=1, padding='SAME')
```

Stride is the number of pixels shifts over the input matrix. In the case above, initializing stride 1 means moving the filter with one pixel at a time. And if we initialize stride 2 then the filter will be moved by 2 pixels.

Another parameter that can be seen is padding. There are two types of padding, valid and same. If padding is the same, then the result of the convolution operation will be as large as the input data. We say that all the pixels in the image will be passed through the convolution operation

Valid padding which keeps only valid part of the image means this operation drop a part of imagine, where the filter did not fit.

Convolution layers are subject to an activation function to ensure nonlinear network behavior. As an activation function, ReLU (Rectified Linear Unit) is often used. Figure 2.12 illustrates the result of applying the ReLU function to a feature map.

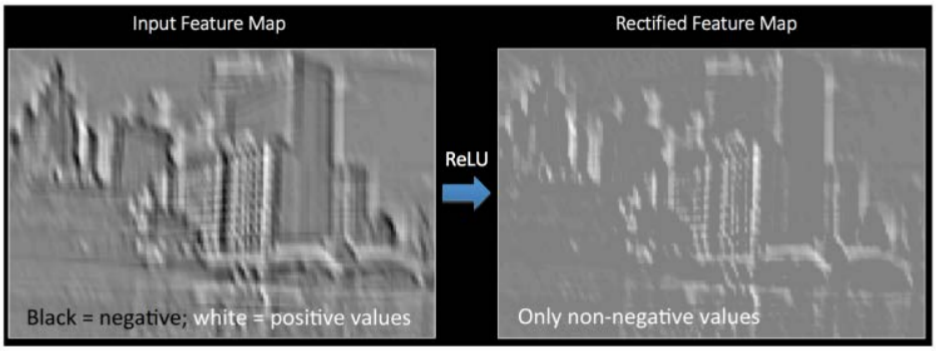


Figure 2.12: Applying Rectified Linear Unit to a feature map

2.2.2 Pooling Layer

Just like strides, pooling is another way of reducing the dimensionality of a layer. Depending on the task, one may choose from different pooling methods. Similar to convolution operation, pooling methods also work with patches and strides.

Max Pooling will calculate the maximum value in a channel within the patch.

Average Pooling will calculate the average value in a channel within the patch.

Sum Pooling will calculate the sum from a channel within the patch.

For exemple in Figure 2.13 we can see how max pooling work. Here we will see three types of pooling layers.

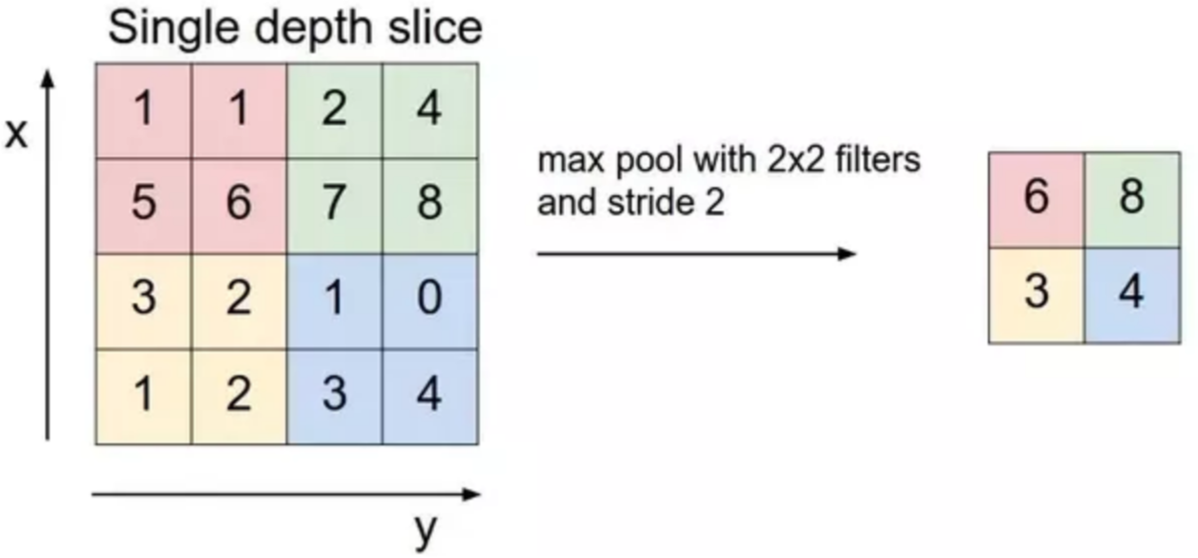


Figure 2.13: Max pooling

2.2.3 Fully Connected Layer

As the name suggests, each node in the layer must be connected to each node so we can say that fully connected is actually like a neural network. Within this layer the classification is made, so as input to this layer are provided feature maps and on the output data the softmax activated function is applied.

The output of this layer is a probability vector with a number of components equal to the number of classes. Each component of the vector represents the probability that the input image will fit into the appropriate class.

In Figure 2.8 you can see this process, the first map matrix will be converted as vector and fully connected layers combined these features together to create a model. After applying the softmax function to get the probabilities.

Chapter 3

Related work

3.1 DeeperCut: A Deeper, Stronger, and Faster Multi-Person Pose Estimation Model [12]

3.1.1 Problem

The goal of this paper is to advance the state-of-the-art of articulated pose estimation in scenes with multiple people.

3.1.2 Methods

Part Affinity Fields (PAFs), to learn to associate body parts with individuals in the image. The architecture encodes global context, allowing a greedy bottom-up parsing step that maintains high accuracy while achieving realtime performance.

The part affinity is a 2D vector field for each limb: for each pixel in the area belonging to a particular limb, a 2D vector encodes the direction that points from one part of the limb to the other. Each type of limb has a corresponding affinity field joining its two associated body parts.

3.1.3 Data

The MPII human multi-person dataset [8] and the COCO 2016 keypoints challenge dataset [16]. These two datasets collect images in diverse scenarios that contain many real-world challenges such as crowding, scale variation, occlusion, and contact.

MPII Human Pose dataset is a state of the art benchmark for evaluation of articulated human pose estimation. The dataset includes around 25K images containing over 40K people with annotated body joints. The images were systematically collected using an established taxonomy of everyday human activities.

3.1.4 Performance and comparisons

Evaluation is done on two single-person and two multi-person pose estimation benchmarks. The proposed approach significantly outperforms best known multi-person pose estimation results while demonstrating competitive performance on the task of single person pose estimation.

3.2 Human Pose Estimation from Monocular Images: A Comprehensive Survey [23]

3.2.1 Problem

In this paper, a comprehensive survey of human pose estimation from monocular images is carried out including milestone works and recent advancements. The goal of our application is to provide initialization for automatic video surveillance.

3.2.2 Methods

Based on one standard pipeline for the solution of computer vision problems, this survey splits the problem into several modules: feature extraction and description, human body models, and modeling methods. There are additional sections for motion-related methods in all modules: motion features, motion models, and motion-based methods.

3.2.3 Data

The paper collects 26 publicly available datasets for validation and provides error measurement methods that are frequently used.

3.2.4 Performance and comparisons

The first survey that includes recent advancements on human pose estimation based on deep learning algorithms. Although deep learning algorithms bring huge success to many computer vision problems, there are no human pose estimation reviews that discuss these works. In this survey, about 20 papers of this category are included. This is not a very large number compared to other problems, but this is a inclusive survey considering the relatively few works addressing this problem.

3.3 PersonLab: Person Pose Estimation and Instance Segmentation with a Bottom-Up, Part-Based, Geometric Embedding Model [19]

3.3.1 Problem

Paper try to solve the task of 2-D pose estimation and instance segmentation of people in multi-person images using an efficient single-shot model based on a box-free bottom-up approach. The study contributes to computer vision applications such as smart photo editing, person and activity recognition, virtual or augmented reality, and robotics.

3.3.2 Methods

Bottom-up approach for pose estimation and segmentation:

- localizing identity-free semantic entities (individual keypoint proposals or semantic person segmentation labels, respectively)
- grouping them into person instances:
    - use a greedy decoding process to group them into instances
    - train network to predict instance-agnostic semantic person segmentation maps
    - for every person pixel we also predict a set vectors to each of the K keypoints of the corresponding person instance. (corresponding vectors fields can be thought as a geometric

embedding representation and induce basins of attraction around each person instance, leading to an efficient association algorithm)

- ∗ For each pixel xi, they predict the locations of all K keypoints for the corresponding person that xi belongs to
- ∗ then compare this to all candidate detected people j (in terms of average keypoint distance), weighted by the keypoint detection probability
- ∗ if this distance is low enough, we assign pixel i to person j

3.3.3 Data

Standard COCO keypoint dataset [16] , which annotates multiple people with 12 body and 5 facial keypoints.

3.3.4 Performance and comparisons

- compared to the best previous bottom-up approach they improve keypoint AP (Average Precision) from 0.655 to 0.687
- compared to the strong top-down FCIS method [15] improve mask AP from 0.417 to 0.386

3.4 Towards Accurate Multi-person Pose Estimation in the Wild [20]

3.4.1 Problem

Paper try to solve the task of multi-person detection and 2D pose estimation based on a top-down approach (2-D localization of human joints on the arms, legs, and keypoints on torso and the face). The study want to localize people, understand the activities they are involved in, understand how people move for the purpose of Virtual/Augmented Reality, and learn from them to teach autonomous systems.

3.4.2 Methods

Top-down approach consisting of two stages:

- first stage, predict the location and scale of boxes which are likely to contain people, use Faster RCNN detector.
- second stage, estimate the keypoints of the person potentially contained in each proposed bounding box:
  - for each keypoint type predict dense heatmaps and offsets using a fully convolutional ResNet
  - combine these outputs by a novel aggregation procedure to obtain highly localized keypoint predictions
    - use keypoint-based Non-Maximum-Suppression (NMS), instead of the cruder boxlevel NMS
    - keypoint-based confidence score estimation, instead of box-level scoring

3.4.3 Data

Training data: standard COCO keypoint dataset [16], which annotates multiple people with 12 body and 5 facial keypoints.

3.4.4 Performance and comparisons

- average precision of 0.649 on the COCO test-dev set and the 0.643 test-standard sets, outperforming the winner of the 2016 COCO keypoints challenge and other recent state-of-art
- using additional in-house labeled data we obtain an even higher average precision of 0.685 on the test-dev set and 0.673 on the test-standard set, more than 5% absolute improvement compared to the previous best performing method on the same dataset

3.5 Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields [9]

3.5.1 Problem

The purpose is to recognize a layout of the body parts, based on the pose estimation and Pose Affinity Fields (PAF)

3.5.2 Methods

- Part affinity fields(PAF): is a set of 2D vector fields that encode the location and orientation of the limbs. It is a set of vectors that encodes the direction from one part of the limb to the other; each limb is considered as an affinity field between body parts.
- Estimation of body-part confidence maps.

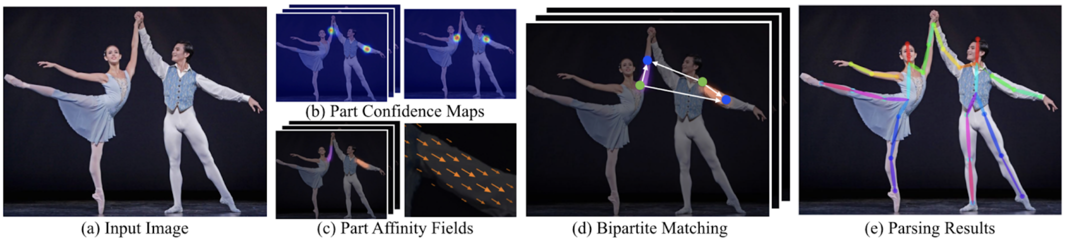


Figure 3.1: Overall pipeline

3.5.3 Data

As an input we have an image. This method simultaneously infers two maps with body-parts, as you can see in b figure and runs a special bipartite matching algorithm. In the end it assemble the body parts into full body poses.

Training data: standard COCO keypoint dataset [16] and the MPII human multi-person dataset [8]

Chapter 4

System architecture and specifications

In this chapter we will begin by describing the objectives of the system presented in this paper. We will provide information on the development of the two prototypes. At the same time, we will highlight the capabilities of the application. Finally, we will discuss possible extensions and results.

4.1 Goals

Our motivation is to help patients who need physiotherapy to see a progress and to encourage them to be constant during their treatment. We want to support patients motivation in continuing to build new and healthy behaviours. But also helping Physiotherapist in evaluation of patient.

The primary objective is to track the patient while he doing exercise to help the physical therapist in evaluation of patient and to motivate the patient by reporting progress.

In this case, we implemented the pose estimation algorithm, the two approaches detailed in the introduction of the papers, the one that calculate Range of Motion and the calculation of the distances for display the progress of the patient

But to achieve this goal, we need the pose estimation algorithm to have a minimum of 30 fps that can run both in the browser and on mobile devices.

Next we will accurately run real-time pose estimation algorithms on the video camera at a performance of at least 30 frames per second. But also finding convolution neural network architectures to allow this.

We do not want to use a server that are to receive the streams of images and apply detection algorithms due to calculations costly in time and volume of data to be processed.

The server variant was implemented but the big problem was a delay of 3-4 sec so our goal of at least 30 frames per second could not be achieved.

To achieve maximum performance of the algorithms, it is necessary to run them on the GPU of the client application.

We proposed the implementation of two prototypes, a web application and a mobile application. Each application implements a different approach that uses a different type of architecture from convolution neural network.

The mobile app will calculate the distance traveled by the hands and legs during an exercise that the patient will perform, generating reports with day-to-day performance to improve motivation of the patient.

The web application will calculate a range of motion to help the patient perform the correct exercises, so we will count the correct number of repetitions of an exercise based on the angles that help the patient during the exercise.

4.2 Mobile - Application development

4.2.1 Specification of the problem

This application is designed to help everyone who need physiotherapy treatment to stay motivated, reach their goals, and create habits that are healthy and helpful for a long term. In this case we have a solution by creating a app which will be a useful tool for patients who needs help to reach their goals by automated tracking of distances.

4.2.2 Analysis and design

Pentru mobile exista tensorflow Lite care este un model antrenat optimizat care permite rularea pe orice telefon. Ideei [24]

- Acesta arhitectura a fost special conceputea pentru pose estimation
- The contribution of this paper is to implicitly model long-range dependencies between variables in structured prediction tasks such as articulated pose estimation.

4.2.3 Implementation

4.2.4 User manual

4.3 Web - Application development

In acest capitol vom descrie modul in care a fost dezvoltat aplicatia suprinzand principale algoritmi folosinti. Continund cu detalile de implementare care au contribuit la acesta aplicatie.

4.3.1 Specification of the problem

Application is an interactive software destined for patients who need physiotherapy treatments. Our application guide patient to see how they need to make their exercises in a correct manner by showing Range of Motion (ROM) in real time and after allowing us to count the number of movements.

It is based on exercices, because we think that a constant and correct number of movements could be more efficient for patients then just to present them what exercices they need to do. In this manner the application guides each patient during the all period they need to follow their treatment.

4.3.2 Analysis and design

Aplicatia va rula in browser folosind camera web pentru a detecta in real-time postura pacientului. Oferindune cordonatele membrelor acestuia pe baza carora vom aplica algoritmul de calculare a unghiurilor, obtinand astfel range of motion. In acest sens vom folosit tensorflow.js care permitea rularea in browser a algoritmului de inteligenta artificiala. Noi vom folosi modelul PoseNet.

PoseNet este un model antrenare care are la baza o reatea neuronale de convolutie, find invatata sa clasifice partile corpului dand ca rezultat cordonatele acestora. In figura 4.1 se poate observa o diagrama care prezinta structura deletor de iesire care le ofera PoseNet.

PoseNet a fost realizat de cei de la google si foloseste o arhitectura de tip MobileNet. MobileNet este o arhitectura optimizata pentru a rula pe dispozitive fara putere mare de calcul.

In articolul " Mobilenets: Efficient convolutional neuralnetworks for mobile vision applications" [11] sunt descrise principale optimizare aduse de arhitectura numit MobileNet. Cea mai importanta optimizare de performanta a fost realizata prin introducerea unui nou tip de strat, numit Depthwise

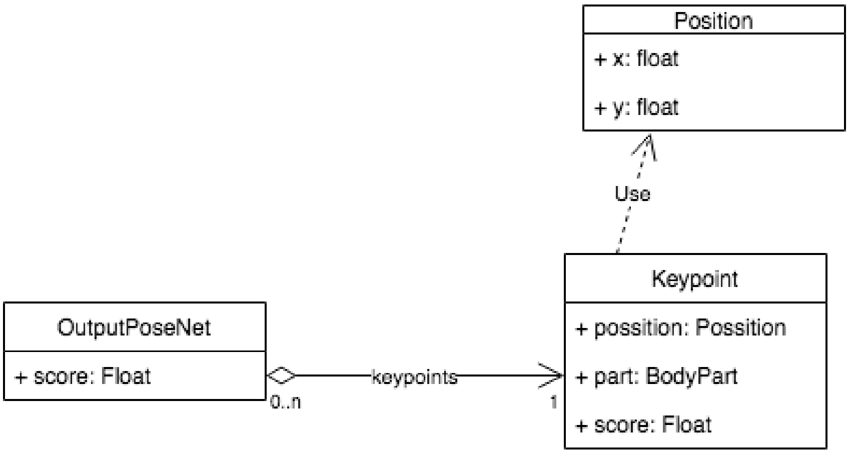
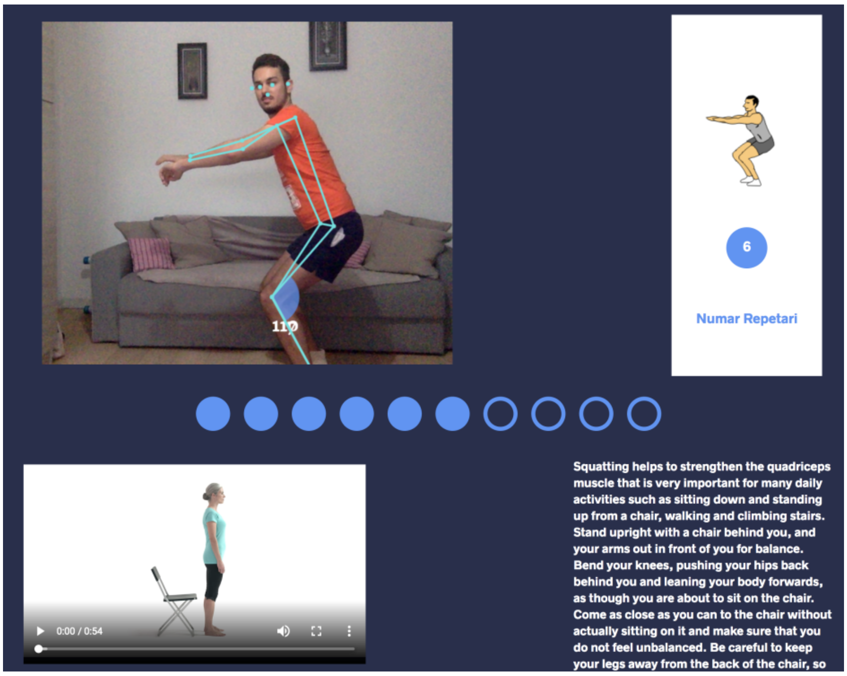


Figure 4.1: PoseNet output

Separable Convolution. Acest strat inlocuieste convolutila standard. Si mare diferenta este ca a impartit in doi pasi separati, aplicarea filtrari si combinarea.



Squatting helps to strengthen the quadriceps muscle that is very important for many daily activities such as sitting down and standing up from a chair, walking and climbing stairs. Stand upright with a chair behind you, and your arms out in front of you for balance. Bend your knees, pushing your hips back behind you and leaning your body forwards, as though you are about to sit on the chair. Come as close as you can to the chair without actually sitting on it and make sure that you do not feel unbalanced. Be careful to keep your legs away from the back of the chair, so

Figure 4.2: Demo MyKineto

Pe baza keypoints obtinute de la PoseNet vom desena unghiurile dintre membrele pacientului folosind canvas. Unghiul dintre doua drepte va fi calculat folosind formula matematica, reprezentand range of motion.

Pentru a numara numarul de exerciti corecte vom folosi un algoritm bazat pe unghiuri pentru fiecare tip de exercitiu.

De exemplu daca se fac squats atunci algoritmul va functiona pe baza de stari, astfel vom avea starile:

- Inceputul exercitiului daca unghiul de la genunchi este intre 10 si 75 de grade
- Sfarsitul exercitiului daca unghiul de la genunchi este intre 145 si 300 de grade

Pe baza acestor doua stari, vom numara numarul de exerciti facute corect din puncte de vedere a unghiului genunchiului. In Figura 4.2 se poate vedea cum algoritmul numara numarul de repetari corecte.

4.3.3 Implementation

Pentru implementarea acestei aplicatie web sa folosit react in dezvoltarea interfetei grafice. Iar pe partea de pose estimation am folosit tensorflow.js impreuna cu modelul PoseNet.

Ce este React? React este o bibliotecă JavaScript folosită în dezvoltarea interfețelor web. Prima versiune a fost lansată de către facebook în Martie 2013. Față de alte platforme React vine cu capacitatea sa de a fi declarativ, bazat pe componente și faptul că este reutilizabil [5].

Biblioteca React permite crearea de componente pentru fiecare stare a aplicației, în cazul modificări unei stări se vor randa componentele care trebuie să apară în cazul acelei stări, astfel se crește performanța aplicației.

React folosește noua sintaxă JavaScript Extension (JSX) care este asemănătoare cu sintaxa Hypertext Markup Language (HTML), defirența este că pe post de marcatori putem folosi componente React [3].

O cauză a modelului declarativ bazat pe componente constă în reutilizarea codului, fiecare componentă își încapsulează și gestionează propria stare. Se pot realiza interfețe complexe prin asamblarea lor precum niște pazăluri. Fiecare componentă își gestionează, asta este un task dificil în realizarea unor arhitecturi complexe.

Pentru o eficiență mai sporită, React folosește Virtual DOM, care este o abstractizare a DOM, fiindcă operațiile asupra DOM sunt încete [2].

Tensorflow.js permite rula algoritmilor de inteligenta artificala in mediul javascript. Dupa cum stim algoritmi de inteligenta artificiala, mai ales cei aplicati pe imagini sau video au nevoie de o putea mare de calcul. Pentru a rezolva acesta problema, tensorflow.js ruleaza algoritmi folosind WebGL in mod automat. Acest lucra face ca algoritmi sa ruleze pe GPU [6].

Pe lângă asta, pentru dezvoltarea aplicație s-a folosit typescript care este un super set tipizat a lui javascript și mediul de dezvoltare Visual Studio Code.

4.4 Experimental results

4.5 Possible extensions

Chapter 5

Conclusion

Proposed system for the tracking of the posture doesn't perform as we originally expected.

There seems to be some problems regarding the determination of the bounding box, maybe due to the fact that the detected points are on a surface that is characterized by just a few colors that appear in almost all tracked region. Because of these characteristics of the tracked posture we fail to track enough points due to insufficient difference in the nearest proximity of a point.

For future development we will try to adapt described system to track the contours.

Even if posture tracking system doesn't perform as we expected, we succeeded in the finding the appropriate settings for the PoseNet so that it performs enough good for the determination of the ranges of motion. Based on these calculations we were able to count, in real time, the number of exercises that user made.

Bibliography

[1] Convolutional neural network - mathworks. https://www.mathworks.com/solutions/deep-learning/convolutional-neural-network.html, May 2019.

[2] The difference between virtual dom and dom - react kung fu. https://reactkungfu.com/2015/10/the-difference-between-virtual-dom-and-dom/, May 2019.

[3] Jsx in depth - react. https://reactjs.org/docs/jsx-in-depth.html/, May 2019.

[4] Neuronal networks - history. https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/History/, May 2019.

[5] React - a javascript library for building user interfaces. https://reactjs.org/, May 2019.

[6] Tensorflow.js. https://www.tensorflow.org/js, May 2019.

[7] Wharton M. A. *Health Care Systems I, Slippery Rock University*. 1991.

[8] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. 2014. In CVPR.

[9] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. *CoRR*, abs/1611.08050, 2016.

[10] Conf. dr. Mircea Ifrim, Prof. dr. doc. Gherghe Niculescu, Prof. dr. N. Bareliuc, and Dr. B. Cerbulescu. *Atlas de anatomie umana, Volumul III*. 1985.

[11] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017.

[12] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. *CoRR*, abs/1605.03170, 2016.

[13] Richard W Keen. Burden of osteoporosis and fractures. *Current Osteoporosis Reports*, 1(2):66–70, 2003.

[14] Ryan Klepps. 7 thought-provoking facts about physical therapy you can't ignore. https://www.webpt.com/blog/post/7-thought-provoking-facts-about-physical-therapy-you-cant-ignore, May 2019.

[15] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. Fully convolutional instance-aware semantic segmentation. *CoRR*, abs/1611.07709, 2016.

[16] Lin, T.Y., Cui, Y., Patterson, Bourdev, Girshick, and Dollr. Coco 2016 keypoint challenge. 2016.

[17] Stphane Mallat. *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*. Academic Press, Inc., Orlando, FL, USA, 3rd edition, 2008.

[18] Peter Norvig and Stuart J. Russell. *Artificial Intelligence - A Modern Approach*. Prentice Hall, 1994.

[19] George Papandreou, Tyler Zhu, Liang-Chieh Chen, Spyros Gidaris, Jonathan Tompson, and Kevin Murphy. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. *CoRR*, abs/1803.08225, 2018.

[20] George Papandreou, Tyler Zhu, Nori Kanazawa, Alexander Toshev, Jonathan Tompson, Chris Bregler, and Kevin P. Murphy. Towards accurate multi-person pose estimation in the wild. *CoRR*, abs/1701.01779, 2017.

[21] Raluca Vasilescu. Un scurt istoric al ia. https://www.cs.cmu.edu/~mihaib/articole/ai/ai-html.html, May 2019.

[22] Computer Vision. *Health Care Systems I, Slippery Rock University*. Prentice Hall, 2001.

[23] Gong W, Zhang X, Gonzàlez J, Sobral A, Bouwmans T, Tu C, and Zahzah EH. Human pose estimation from monocular images: A comprehensive survey. *PubMed*, PMID: 27898003, 2016. doi: 10.3390/s16121966.

[24] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. *CoRR*, abs/1602.00134, 2016.