N^o d'ordre: XX/2024

THESIS SUBMITTED TO THE

UNIVERSITY OF MOHAMED BOUDIAF – MSILA

FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
DEPARTMENT OF COMPUTER SCIENCE

MASTER THESIS

In partial fulfillment of the requirements for the degree of

Master in : Computer science

Specialty : Artificial Intelligence

By :

BENAISSA AYA

BENAISSA DOUNIA HANAA

Title of the thesis

COMPUTER VISION FOR BODY WEIGHT ESTIMATION

Under the supervision of

Mr.Abdessattar Ghemougui

Composition of the jury

M.	**Brahimi Belkacem**	University of Msila	President
M.	**Said Hamani**	University of Msila	Examiner

2023/2024

the one who helped me reach this rank, to the one who enlightened my path with advice of success and steadfastness, and who was a symbol of determination, to my dear mother, may God prolong her life And sHe made it the best remembrance on the Day of Judgment. To the one who was my support and whom I see as a symbol in this life, my father, and to all my dear brothers, each in his name. To all my loved ones... To those who supervised the completion of this work of mine **(Mr.Abdessattar Ghemougui)**. To everyone who reads this work, I dedicate this achievement, hoping that it will resonate widely in insight, insight, and action.

Benaissa Aya

To my parents: I cannot repay you with anything for what you have done for me all this life. But still, I am trying to repay you by saying 'Thank You' for everything.
To my family: It is only because of you, my family, that I have been able to live such a wonderful life. Thank you for being there every step of the way, from my childhood through adulthood.
To my friends: Rihab ,Maroi,Houda,Soumia,Halima,Yasmine Aziza thank you for being in my life I am gratefull for your freindship.
lasty My dear teacher **ELBAHI SAEDIA** am lucky to be your student and I want to say thank you for everything

Benaissa Dounia Hanaa

Acknowledgement

we want to express our deepest gratitude and appreciation to the following individuals and institutions that have contributed to the completion of this thesis.

First and foremost, we are immensely thankful to my thesis advisor,**Dr.Abdessattar Ghemougui** . Their guidance, expertise, and continuous support have been invaluable throughout this research journey. Their patience, constructive feedback, and unwavering commitment to my academic development have been instrumental in shaping this thesis. we are truly grateful for their mentorship. we would also like to extend our gratitude to the faculty members of the Informatics Department at Msila University, whose knowledge and expertise have greatly influenced our intellectual growth. Their dedication to teaching and research has been a constant source of inspiration.

We would also like to express our appreciation to the participants who willingly took part in this study. Their cooperation and willingness to share their experiences and insights have enriched the findings of this thesis.

Lastly, we would like to thank all those who have directly or indirectly contributed to this thesis. Your contributions, whether through discussions, feedback, or encouragement, have played a significant role in shaping the outcome of this research. To everyone mentioned above and to those whose names might have inadvertently been omitted, please accept my sincere thanks. Your support and encouragement have been crucial in making this thesis a reality.

ملخص

تقدير الوزن القائم على الرؤية الحاسوبية هو مجال بحثي يهدف إلى تقدير وزن جسم أو حيوان أو إنسان باستخدام تقنيات الرؤية الحاسوبية. تستخدم هذه الطريقة خوارزميات معالجة الصور وأساليب إعادة البناء ثلاثية الأبعاد لاستخراج السمات ذات الصلة من الصور وإجراء التنبؤ بالوزن باستخدام الشبكة العصبية التلافيفية والتعلم التحوّلي.

الكلمات المفتاحية: الشبكة العصبية التلافيفية،تقدير الوزن، الرؤية الحاسوبية، التعلم الآلي، التعلم العميق،التعلم التحوّلي

Abstract

Computer vision-based weight estimation is a field of research that aims to estimate the weight of an object, animal, or human using computer vision techniques. This method uses image processing algorithms and 3d reconstruction methods to extract relevant features from images and perform weight prediction using a convolutional neural network and transfer learning.

Keywords: Weight estimation, Computer vision, Machine learning, Deep learning, CNN, Transfer Learning.

Abstract

L'estimation du poids basée sur la vision par ordinateur est un domaine de recherche qui vise à estimer le poids d'un objet, d'un animal ou d'un être humain à l'aide de techniques de vision par ordinateur. Cette méthode utilise des algorithmes de traitement d'image et des méthodes de reconstruction 3D pour extraire les caractéristiques pertinentes des images et effectuer la prédiction du poids à l'aide d'un réseau neuronal convolutionnel et de l'apprentissage par transfert.

Mots Clée: Estimation du poids, vision par ordinateur, apprentissage automatique, apprentissage profond,réseau neuronal convolutionnel,l'apprentissage par transfert.

Contents

List of Figures

List of Tables

General Introduction

General Introduction

The integration of artificial intelligence (AI) has greatly accelerated the evolution of various domains, including weight estimation. This estimation of weight is crucial for understanding the health of humans and animals, as it serves as a significant indicator of overall well-being and helps identify normal or abnormal changes in health status. However, traditional methods like manual or digital scales have limitations in accurately estimating weight.

In the past, weight estimation relied on conventional methods that involved direct physical contact with traditional scales. However, with the advancement of computer vision techniques, weight estimation has undergone a significant transformation. This evolution has shifted weight estimation from traditional methods to the utilization of computer vision techniques.

Computer vision-based weight estimation is a cutting-edge approach that leverages machine learning algorithms, image processing, and regression methods. By harnessing the power of these technologies, weight estimation can now be conducted without the need for direct physical contact with the object or subject in question. This breakthrough in weight estimation allows for more accurate measurements and enhances efficiency in caring for the health of individuals and animals.

0.1 Statement of the problem

AI technologies, particularly age and gender recognition, have made significant advancements. However, the same level of progress has not been observed in weight estimation compared to age and gender recognition. The development of AI algorithms for weight estimation has not kept pace with the advancements in other areas. This discrepancy highlights the need for further research and improvement in weight estimation algorithms to achieve a more balanced development across different AI applications. Our study sheds light on weight estimation using computer vision techniques. During our research, we encountered several problems, including:

1. What are the challenges in estimating weight using computer vision techniques?

2. Is it possible to estimate animals and human weight using computer vision techniques?

3. Which algorithm is used for weight prediction in computer-based systems?

0.2 Objective

Our weight estimation project aims to revolutionize the process of obtaining real-time weight readings for people, saving valuable time, effort, and money. Instead of relying on visits to pharmacies or medical centers, individuals can now leverage image estimation technology for easy and efficient weight measurements.

This innovative solution offers a convenient and cost-effective alternative for individuals, particularly the elderly, who may find it inconvenient or costly to visit medical facilities for weight readings. By providing a user-friendly and accessible method of weighing, this project addresses the needs of individuals who may face challenges in accessing traditional weighing services.

0.3 Report outline

This document is divided into three chapters plus the conclusion as follows:

Chapter 1: Weight estimation

• The First chapter presents a general introduction to computer vision-based weight estimation we delve into the wide range of applications, advantages, challenges, and devices associated with this approach.

Chapter 2: State of the art of weight estimation

• The second chapter delves into the current advancements in weight estimation, in the animal field and human field. At the end of this chapter, we highlight the comparisons between animal and human weight estimation techniques.

Chapter 3: Implementation and Experimental Results

• The third chapter explains selecting human weight estimation as the focus of our project. It also focuses on the implementation and experimental results of the project

General Conclusion

Finally, we offer a conclusion of our work, encompassing the key aspects of our project, including its advantages, limitations, and future possibilities.

Chapter 1
Weight Estimation

Chapter 1

Weight estimation

1.1 Introduction

In this chapter, we will begin by examining the various methods employed in computer vision-based weight estimation. Additionally, we will delve into the wide range of applications, advantages, challenges, and devices associated with this approach. Doing so will give us a comprehensive understanding of the subject matter at hand.

In recent years, the development of AI applications has experienced exponential growth fueled by technological advancement and increased adoption across industries. The value of the AI market is expected to grow 20-fold by 2030, reaching $2 trillion U.S dollars . This growth is driven by the adoption of AI across industries, including chains, marketing, research, and analysis. Weight estimation is one of the methods that has undergone an evolution from traditional to the use of computer vision techniques

1.2 Computer vision-based weight estimation

Weight is typically measured in units like kilograms or pounds. It can be used in a wide range of fields, including livestock, healthcare, agriculture, transportation, sports, fitness, and the chemical industry.

Computer vision-based weight estimation utilizes image processing, machine learning algorithms, and regression methods to estimate the weight of objects or subjects without direct physical contact. By analyzing visual data, such as images or videos, and extracting relevant features, a computer vision model can be trained to make predictions about the weight based on the extracted features.

1.3 Computer vision-based weight estimation applications

Computer vision-based weight estimation has found significant applications in various fields,here are some key applications:

1.3.1 Livestock Weight Estimation

Due to the rapid growth of the cattle and pigs that provide meat products to the growing world population [?, 9–12]the monitoring of the weight of cattle and pigs has become more important [13]. Computer vision-based weight estimation has increased because it helps to monitor the health and growth of animals, and helps farmers or veterinarians to identify health issues or nutritional deficiencies early and take appropriate actions to address them.

Compared to traditional weight estimation methods, this can save time and also improve animal welfare, prevent diseases, and increase the quality of dairy products and meat. In figure1 computer vision reduces the need for manual handling, thus reducing the requirement for human intervention, labor requirements, and minimizing stress on the animals..

Figure 1. Three Kinect v2 cameras installed at the left, right and top positions [13]. However, there are some safety concerns due to the narrow road railings [14] developed an arch device with five Kinect DK cameras located at the top, upper left, lower left, upper right, and lower right. For the safety of equipment and personnel, the railing in the center of the arch is thickened, but due to the obstruction of the railing, data collection is affected to a certain extent.

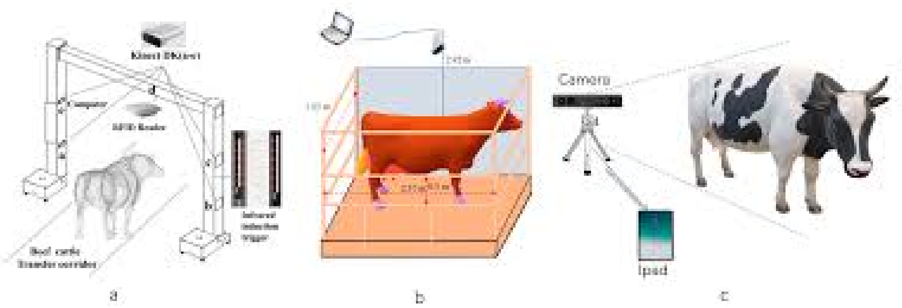


Figure 1.1: Computer vision based-weight estimation in livestock (a). Canal elbow type , (b). Suspension fixed type , (c). Simple and portable [1].

1.3.2 Health Care

Computer vision-based weight estimation has already been applied to the medical domain and body analysis. Algorithms like silhouette analysis [15]. Automatic extraction of the weight [16]and measures of the body [17], as well as a 3D model reconstruction of the human body [18], may be used as a self-diagnostic tool for telemedicine.

Computer vision-based weight estimation is often necessary for medical treatments, such as the emergency treatment of stroke patients: Figure 2 shows the case of ischemic stroke, the patient has to receive a body weight-adapted drug, to solve a blood clot in a vessel. The accuracy of the estimated weight influences the outcome of the therapy directly. However, the treatment has to start as early as possible after the arrival at a trauma room, to provide sufficient treatment. A patient receiving a too-low dose has an increased risk that the blood clot does not dissolve and brain tissue is permanently damaged and a patient receiving an overdose can cause bleeding and further complications [19].

Figure 2 [2] shows a scene with the patient on a stretcher and the complete system in the trauma room as described by Pfitzner et al show the setup with the patient lying on a medical stretcher integrated into the ceiling.

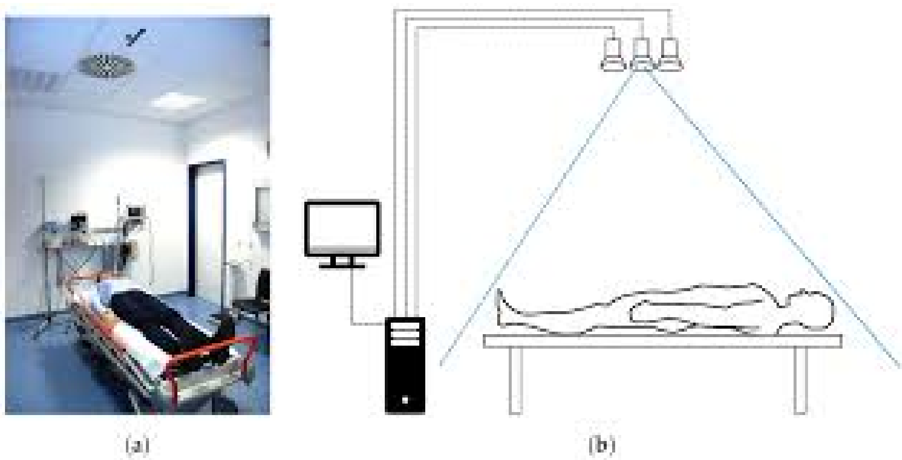


Figure 1.2: Patient weight estimation. [2]

1.3.3 Agriculture and Food Processing

.In agriculture Computer vision-based weight estimation is widely used for quality control and inspection of products [20]. Their applications about fruit and vegetables(figure

3) [3] defect detection such as common defect detection on citrus [21]. size assessment of berries [22], size estimation of sweet onions [23, 24].

The size and weight of poultry meat and eggs are essential for production economics. Computer vision-based weight estimation has become a promising tool in the real-time automation of poultry weighing and processing systems.

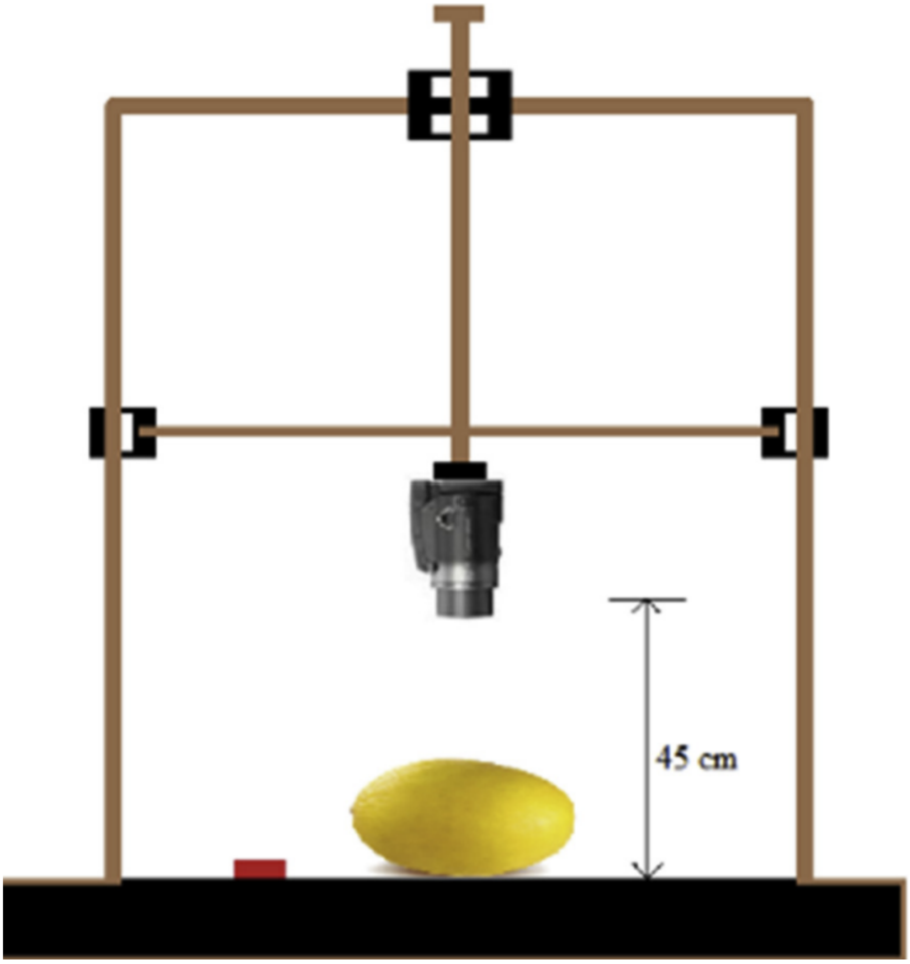


Figure 1.3: Computer vision system for estimating the weight of lemons. [3]

1.3.4 Transportation

Cargo trucks are used when it is necessary to transport goods and materials with dimensions or weights exceeding the standard in traffic regulations.

Compared to regular trucks, exceptional vehicles with cargo are very heavy, even hundreds of tons, which increases the potential for damage to infrastructure such as bridges and roads. The statistic weighing method uses an electronic scale for weight measurement, which is not suitable for cargo trucks that are much longer than the weighing scale.

Computer vision-based weight estimation techniques are widely utilized in transportation because they would develop low-cost and non-contact measurement and simple installation associated with computer vision techniques.

Figure 4 [4] shows one of the most important applications of computer vision for estimating the weight of a moving vehicle.

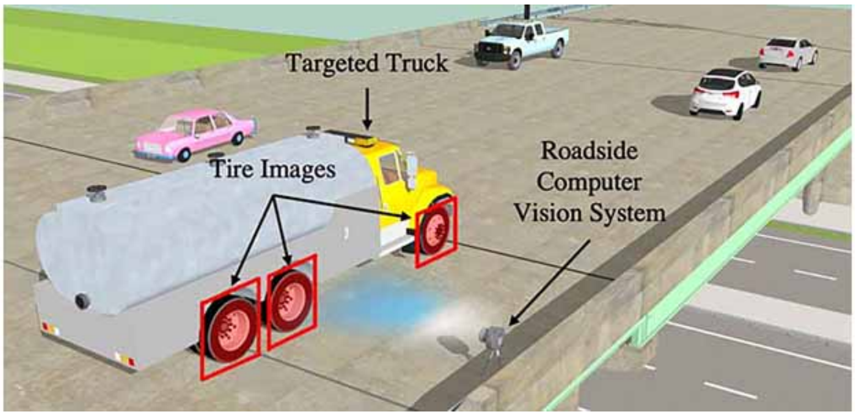


Figure 1.4: Application of computer vision for estimating the weight of moving vehicle [4].

1.3.5 Sports and fitness

In The world's development of economy and the improvement of living standards(lifestyle), people pay more and more attention to the improvement of their overall physical fitness.

Computer vision-based weight estimation has emerged as a valuable tool in the sport and fitness sector. It provides data for performance analysis to track changes in the athlete's body's weight over time, allowing coaches and trainers to make informed decisions about training programs and nutrition plans and achieving fitness goals.

1.3.6 Chemical industry

The number of raw materials needed for production can be determined by weight estimation, also calculating the number of finished products, and ensuring accurate measurements for safety and regulatory compliance. the high accuracy of weight estimation can prevent accidents and environmental hazards.

1.4 The advantages of Computer vision-based weight estimation

1.4.1 Non-invasive

Computer vision-based weight estimation is non-invasive and does not require physical contact with the object being measured, making it suitable for a wide range of applications such as livestock. This non-invasive approach reduces the stress and potential harm to animals. By minimizing the animals' exposure to humans during the weighing process, the risk of contamination and stress is significantly reduced.

1.4.2 Versatility

Computer vision-based weight estimation can be applied to a wide range of objects and materials, making it a versatile tool for various industries and applications.

1.4.3 Real-Time Monitoring:

Computer vision-based-weight estimation can provide real-time monitoring of objects as they move through a production line or a warehouse, allowing for immediate adjustment or intervention if needed such as monitoring the group of fruits, vegetables, and livestock [?, 25–27]

1.4.4 Data Analysis

Computer vision-based weight estimation generates valuable data that can be used for analysis and optimization of processes, such as identifying trends, patterns, or anomalies.

1.4.5 Time-Saving

Computer vision-based-weight estimation offers time-saving across various sectors.In livestock management, it reduces the time traditionally spent on weighing cattle, which is a labor-intensive and time-consuming process [28].In health care, this technology can save time by automating weight estimation allowing for faster measurement.

1.5 Challenges of Computer vision-based weight estimation

When trying to estimate weight from images, we face several technical challenges, including:

1.5.1 Contrast lights and shadows:

Differences in the presence of shadows and lighting in images may lead to distortions in images and make it hard to determine details accurately.

1.5.2 Image distortion

Computer vision-based weight estimation relies heavily on the quality and accuracy of image data. Image distortion is a significant factor that can affect the performance of such systems as geometric distortion. Geometric distortion such as barrel distortion. pincushion distortion, and perspective distortion, can alter the perceived dimensions of objects in images, leading to inaccurate weight estimations [29,30]. These distortions can be caused by lens imperfection, camera angle, or movement, and they can significantly impact the quality of images used for weight estimation.

1.5.3 Different image processing

Different image processing have a profound impact on the accuracy and reliability of Computer vision-based weight estimation such as:

1.5.3.1 3D Imaging techniques

1. **Point Cloud Limitations**

   Variations in lightning conditions, especially in farm environments, can lead to incomplete point cloud information, which limits the accuracy of weight estimation [31].

2. **Data Capture Constraints**

Livestock must remain still during data capture to avoid non-rigid deformation that cloud severely affects the reconstruction process. This requirement can limit the scalability of point cloud acquisition methods [32].

1.5.4 Lack of a unified reference

In some cases, it may be difficult to obtain uniform standards for measurement, which leads to:

1.5.4.1 Inconsistencies in Research Results

The research community faces significant inconsistencies in results, with average relative errors and validation outcomes [33]. This indicates that without a unified reference, it is challenging to access the precision and reliability of weight estimation methods.

1.5.4.2 Challenges in Body Dimension Measurement:

Different technical methods such as image processing, deep learning, and data mining are used in body dimension measurement, each contributing to the lack of a standardized approach. These varying methodologies lead to different measurement outcomes, which can affect the accuracy of weight estimation.

1.5.4.3 3D Point Cloud Data Analysis

In the analysis of 3D point cloud data, the discrepancies in maximum and minimum errors highlight the inconsistency in measurement accuracy [33]. This inconsistency can lead to errors in weight estimation, as the 3D models used to calculate weight may not be accurate.

1.6 Computer vision-based weight estimation methods

Over the years, weight estimation methods have evolved significantly, from simple visual assessments to more precise and scientific techniques. Initially, weight estimation was done manually by scales. However, with advancements in technology and the availability of high-resolution cameras, computer vision algorithms have been developed to estimate the weight based on visual data.

1.6.1 Computer vision-based weight estimation Devices

1.6.1.1 Weight Estimation Based on Anthropometric Features

The use of anthropometric features, such as lengths and circumferences of body parts. There are serval methods one anthropometric features such as:

1. **Body Mass Index(BMI)**

   which is calculated by dividing a person's weight in kilograms by their height in meters squared.
   Computer vision can be used to estimate body mass index from facial images. Study shows regional global average pooling, deep features and training datasets based on gender and race improve the accuracy of estimating BMI from facial images [34].

2. **Single image :**

   Computer vision-based weight estimation of single images has been studied in:

   (1) **Image Analysis**

   Image analysis techniques have been used for weight estimation of various agricultural materials. For example, image processing techniques have been used to estimate the weight and volume of wheat damage such as sonic, damaged, shrunken, and broken grains. To accurately classify these refractions, color and texture features were extracted [35].

3. **Facial images**

   Weight can be predicted from face images only and that can happen with the extraction of geometric features. The alignment is based on the detected eye coordinates [36].

4. **2D images:**

   A new method for estimating human body weight from 2D images only is proposed, leveraging facial orientation and body joint measurements in deep learning and XGBoost regression models[36] . The method is faster than previous methods due to lower complexity, with facial models outperforming full-body models [37].

5. **3D scanning**

   Body measurement using 3D surface scanning technologies is faster and more convenient and that works with points cloud and depth map.

Research and development of a near-infrared line array structured light vision system that captures 3D reconstruction and height map images of Apple . To build predictive models, image features are extracted from 3D reconstructed images . LS-SVM model with combined features outperforms the PLS model in predicting volume and weight[37]

1.6.2 LIDAR and Photogrammetry

LIDAR(light Detection and Ranging)is a remote sensing technology that uses laser beams to measure distance to objects and create detailed 3D maps of the Earth's surface wich figure 5 shows that [5].LIDAR systems typically consist of a laser source, a scanner, and a GPS receiver, mounted on an aircraft or a drone. The technology emits laser pulses toward the ground, and the reflected light is measured to calculate the distance between the sensor and the Earth's surface.

By analyzing the LIDAR data, computer vision can estimate the volume of an object, which can calculate its weight based on known density values.

Photogrammetry: is the science of obtaining measurements and creating 3D models from photographs. The technology involves capturing a series of overlapping ariel images, which are then processed using specialized software to generate a digital surface model(DSM) or an orthomosaic map. Photogrammetry relies on the principles of triangulation and perspective geometry(volume) to determine the position of objects in the images and produce accurate 3D models. computer vision algorithms can estimate its volume and subsequently calculate its weight.

Both lidar and photogrammetry can be used in combination with machine learning algorithms to improve the accuracy of weight estimation. By training a model on a dataset of known weights and corresponding sensor data, it is possible to develop a predictive model that can estimate weights based on new sensor inputs.

1.6.3 Estimation with Optical Sensors

Weight estimation with optical sensors in computer vision involves using cameras and optical sensors to analyze the size, shape, and movement of objects to estimate their weight. it uses many approaches such as:

1. Use machine learning algorithms to analyze the visual data captured by sensors. These algorithms can be trained on a dataset of images and the corresponding weight of objects to learn patterns and relationships between visual features and weight.

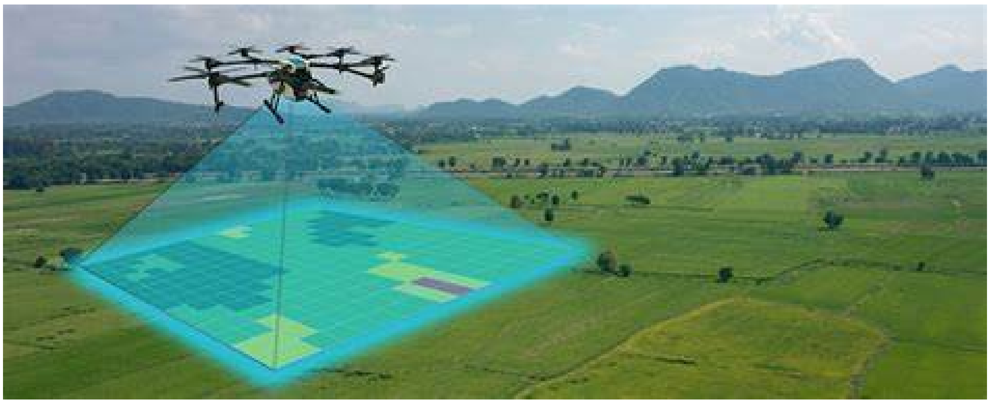


Figure 1.5: Illustration of Lidar and Photogrammetry Technologies [5].

2. use of depth-sensing cameras or other optical sensors that can provide information about the size and shape of an object. By combining this information with known density values for different materials, it is possible to estimate the weight of an object based on its volume and composition.

1.7 Conclusion

In this chapter we thoroughly explore the applications, advantages, challenges, methods, and devices associated with computer vision-based weight estimation. The upcoming chapter will delve into the related work of this approach, providing further insights and deepening our understanding of the subject matter.

Chapter 2
State of the art of weight estimation

Chapter 2

State of the art of weight estimation

2.1 Introduction

In this chapter, we will present a concise overview of the development process involved in both animal and human weighing. Furthermore, we will conduct a comparative analysis of the methods used in computer vision-based weight estimation for animals and humans. By examining these approaches side by side, we can gain valuable insights into the differences between the two domains.

2.2 Computer vision-based animals weight estimation

2.2.0.1 The development process of animal weighing:

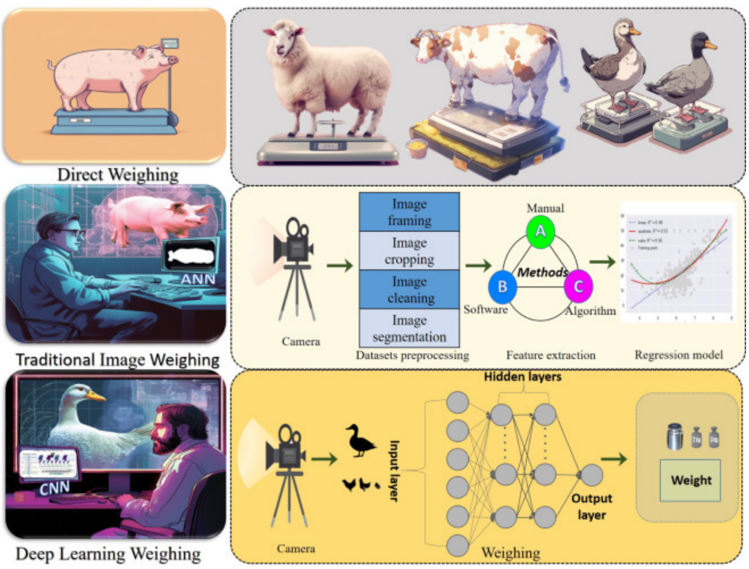


Figure 2.1: The development process of animal weighing [6].

2.2.0.2 Direct Weighing

Direct measurement involves obtaining weight data by placing the animal on a weighing scale and recording the numerical value displayed on the monitor. While this method permits precise of an animal's weight, it necessitates distinct scales for animals with varying body types.

Direct weighing of large animals, such as pigs and cows, demands substantial labor, leading to increased costs or significant stress on the animals. in some unfortunate instances, this approach results in injuries and reduced productivity, while also posing the workers, ultimately creating outweigh the benefits [38–40] .

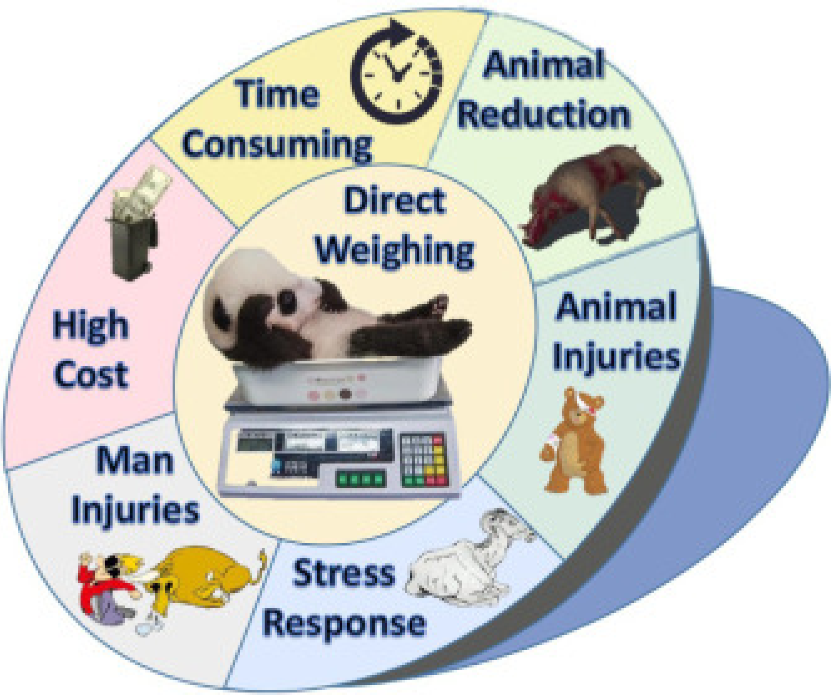


Figure 2.2: Characteristics of animals direct weighing [6].

2.2.0.3 Indirect Weighing

1. **Traditional Image Weighing**

   Compared to direct weighing methods for determining animal weight, traditional image-based weighing techniques offer a non-contact approach.

   This process involves manually measuring a set of body feature parameters from the animal's body using a measuring tape to extract body feature parameters from collected animal images using algorithms approaches or software. Then a feature parameter selection process identifies the parameters with a higher correlation to animal weight.

   These selected parameters are used in mathematical regression methods such as linear regression [8], nonlinear regression, and machine learning to estimate animal weight.

2. **Deep Learning Weighing:**

   Specific animal images or video footage are used as input for the deep learning model. The model outputs the animal's weight.

   Compared to traditional image-based weighing methods, weight measurement methods based on deep learning exhibit a higher degree of automation. It can automatically learn and extract features from images, eliminating the need for manual feature design and selection, thereby enhancing the automation of the weight estimation process and reducing the requirement for human intervention.

   Additionally, it exhibits enhanced generalization abilities, and deep learning models often display the capability to adapt to different species and various animal environments, whereas traditional computer vision methods may necessitate species-specific feature engineering for better addressing diversity.

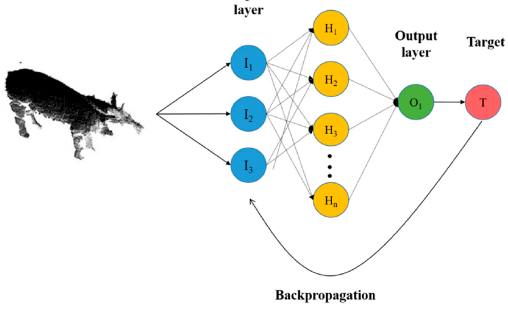


Figure 2.3: Animal Weight Estimation Using Deep Learning [7].

Animals	Weighing Significance	Weighing methods	Characteristics of methods
• Pigs	• Measure growth rate, control fat level, predict market dates, detect disease outbreaks, assess feed efficiency.	• Direct Weighing (Frost et al., 1997; Faucitano and Goumon • Traditional Image Weighing (Schofield, 1990) Schofield et al., 1999) • Deep Learning Weighing (Meckbach et al., 2021)	• Time-consuming, animal injuries, stress response • Frequent monitoring, accuracy is acceptable, less labor input. • Automatically extract features .
• Cattle	• Assess feed efficiency, determine pharmaceutical doses, detect disease outbreaks, economic traits.	• Direct Weighing (Heinrichs et al., 1992; Enevoldsen and Kristensen, 1997; Dingwell et al., 2006) • Traditional Image Weighing (Ozkaya and Bozkurt, 2008) • Deep Learning Weighing (Gjergji et al., 2020)	• time-consuming and costly. • Independent of weighing equipment. • Improved accuracy, end-to-end weighing.
• Broiler	• Control fat level, predict market dates.	• Traditional Image Weighing (Ross and Davis, 1990; Mollah et al., 2010)	• Data collection and analysis time was significantly reduced.

• Ducks	• Economic traits, control fat level	• Traditional Image Weighing (Ogah et al., 2011; Lieng and Sangpradit, 2020) • Deep Learning Weighing (Chen et al., 2023)	• Non-contact, fast, and labor-saving. Automatically extract features, low data analysis costs, end-to-end.
• Fish	• A way to classify fish.	• Traditional Image Weighing (Balaban et al., 2010a,b)	• Repeatability, objectivity, speed, and record-keeping capabilities.

Tableau 2.1: Presents the application scenarios of various standard animal weight parameters, the development process of weighing methods, and their corresponding characteristics [6].

Breed	Device	Method	Animal Numbers	Year
Live pigs	RGB camera	Binocular stero vision technology	32	2004
Live cows	LiDAR	Statistical outlier and voxel grid filtering methods	3	2018
Live cows	LiDAR	Fusion	30	2019
Live pigs	3D camera	Point cloud regisration	20	2018
Newborn lambs	RGB camera	Digital image processing	158	2015
Live sheep	RGB camera	Bionocular stereo vision technology	27	2014
Live cows	LiDAR	Image fusion	25	2023
Live pigs	3D camera	Point coud registration	78	2018
Live cows	3D camera	Point cloud registration	101	2016
Live pigs	Visible image and infrared image sensor	Multi-source image fusion	N/A	2020

Tableau 2.2: Overview of main 3D reconstruction research in Livestock [?]

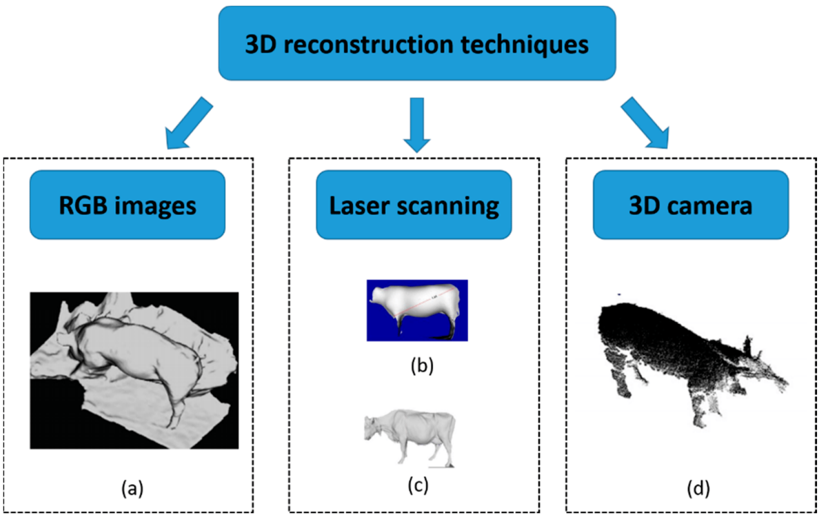


Figure 2.4: 3D reconstruction techniques utilized to estimate the weight of animals [7].

2.3 Computer vision-based human weight estimation

2.3.0.1 The development process of human weighing.

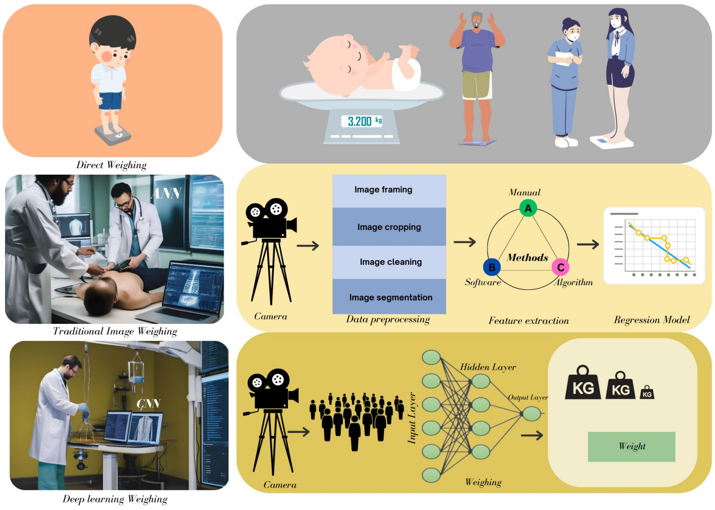


Figure 2.5: The development process of animal weighing

2.3.0.2 Direct Weighing:

Traditional methods of measuring human weight include the use of a weighing scale. The individual places their feet on the scale and reads the displayed weight on the screen. Standard weights such as stone or iron weights, traditionally used for comparison, may also be utilized. This method is commonly used in homes and medical facilities for measuring body weight.

Traditional scales are essential for weighing, but they can have some drawbacks in terms of time and cost consumption.For people who weigh regularly, using a traditional scale may seem time-consuming also the price of a traditional scale itself is expensive and the price varies depending on the type and quality of the scale.

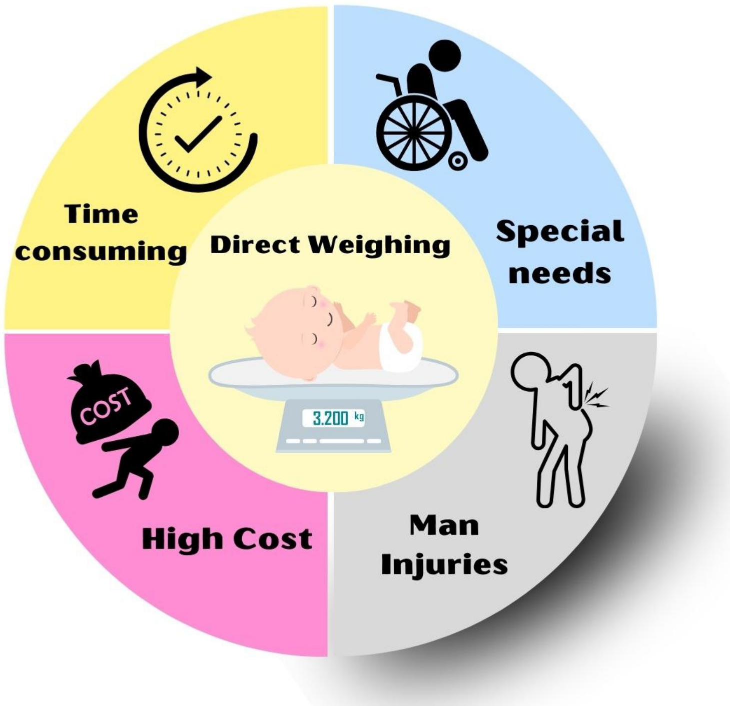


Figure 2.6: Characteristics of human direct weighing.

2.3.0.3 Indirect Weighing

1. **Traditional Image Weighing:**

   Traditional image-based weighing uses visual images to estimate a person's weight. This method typically involves analyzing images of people captured by cameras or other imaging devices to assess their weight or body composition.

   Methods and approaches to extracting features and properties from images using image processing techniques may include extracting measurements of faces, bodies, lines, or other visual cues that may be used.

2. **Deep learning**

   Deep learning-based weighing methods leverage neural network architectures to estimate human weight or body composition using various inputs such as images, sensor data, or biometric measurements.

   These methods have gained popularity owing to their capability to discern intricate patterns and relationships in data, resulting in more precise predictions compared to traditional methods.

   By training deep learning models to analyze images of individuals, it becomes possible to estimate their weight or body composition. Convolutional neural networks (CNN) are frequently employed for this purpose due to their capacity to extract features hierarchically from visual data.

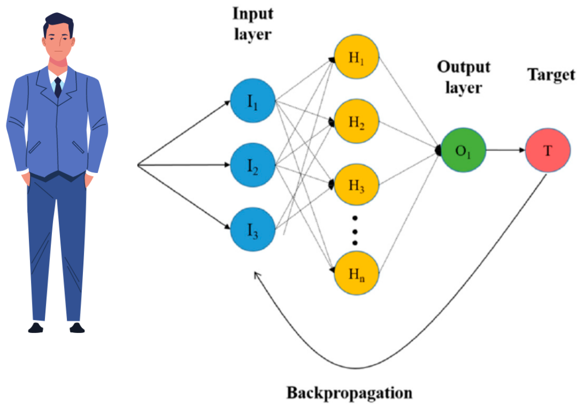


Figure 2.7: Human Weight Estimation with Deep Learning.

Features	Datasets	Feature type(Extraction model)	Classification/ Regression module	Results
Face	MORPH-II(Black and Caucasian)	PIGF(ASM)	SVR	Overall MAE: [3.12– 3.14]
Face	VisualBMI(Caucasian)	Deep features (VGGFace, VGG)	SVR	Pearson correlation: 0.65, 0.47
Face	VIP-attributes (Caucasian)	Deep features(ResNet-50)	end-to-end	Overall MAE :2.36
Face	In-house dataset(Caucasian)	PIGF(ASM)	SVR	Overalll Accuracy: 58.4%
Face	FIW-BMI(Caucasian) MORPH-II(Black and Caucasian)	PIGF,PF,PIGF+PF (Openface) Deep features (VGGFace,LightCNN-29,Centerloss,Arcface)	SVR	Overll MAE Morph-II:[2.30-+0.03-3,77-+0,08] FIW-BMI:[3,15-+0.07-4.26-+0,08]
Face	FIW-BMI(Caucasian) MORPH-II(Black and Caucasian)VIP-attributes (Caucasian)	PIGF,PF,PIGF+PF (Openface) Deep features (Center-loss)	SVR PCA-SVR,GPR, CCA,PLS,LD-PLS	Best Overall MAE Morph-II:(LD-CCA)–2.42 VIPattribute:(LD-CCA)-2.23
Face	Visual BMI (Caucasian) VIP attributes(Caucasian) Bollywood dataset(Indian)	Deep features (ResNet-50,LightCNN-29 MobileNet-V2,VGG-19,DenseNet-121)	SVR,RR	Overall MAE:[1.04,6.48]
Face	Visual BMI(Caucasian) VIP attributes(Caucasian) Bollywood dataset(Indian)	Deep features (VGGFace,FaceNet)	Three Layer(512,256,1) regression module	Overall MAE:[0.32-5.03]
Body	Visual-Body TO BMI DATA Social Net Works	RGB-D NHANES VGG-Net VGG-Face CRE RNN CSJ	SVR SVM GPR CVC	Overall MAE :[0.60-3.00]
Body	Haar cascade	2D ,3D ,image	SVM	Overall MAE |0.24,5,86|

Tableau 2.3: Summary of prior studies on facial analysis-based bmi prediction models in terms of the dataset, machine/deep learning models used, and the obtained error on bmi prediction and obesity classification [8].

2.4 The difference between animals and humans in estimating weight

We can utilize the following table to summarize the differentiation between animal and human weight estimation using computer vision:

Type	Dataset	Article	Devices
Animal	Limited availability of comprehensive datasets	Abundant literature due to extensive	specific cameras
Human	Abundant availability of diverse datasets	sparse literature due to limited data availability	Portable tech devices

Tableau 2.4: The difference between animals and humans in estimating weight

2.5 Conclusion

In conclusion, computer vision plays a pivotal role in estimating the weight of both animals and humans, each presenting unique challenges and considerations. For animals, weight estimation relies on analyzing visual cues like body size and shape, necessitating datasets of diverse species, breeds, and sizes.

Specialized techniques account for anatomical and behavioral differences among species. In contrast, human weight estimation incorporates body measurements and anthropometric data from images or videos, along with contextual factors like age and gender, to enhance accuracy. Hardware requirements vary based on the project's scale and scope. Animal weight estimation often requires specialized equipment for data capture in diverse environments, while human weight estimation can utilize conventional imaging devices in controlled settings.

Chapter 3
Weight estimation using convolution neural networks

Chapter 3

Weight estimation using convolution neural networks

3.1 Introduction

The present chapter assumes a pivotal role within our project, serving as a vital link between the theoretical underpinnings presented in preceding chapters and the practical implementation of those concepts. In this chapter, we will delve into our objective the intricacies of our dataset, model, and corresponding results. By thoroughly examining these components, we aim to provide a comprehensive understanding of the practical application and outcomes of our work.

In the previous chapter, we discussed the challenges associated with implementing computer vision-based animal weight estimation, primarily due to the requirement for specialized and costly equipment like 3D cameras, RGB cameras, and lidar devices. In light of these obstacles, we have decided to redirect our attention toward estimating human weight, which does not rely on using such specialized devices.

3.1.1 objective

Our main aim is to employ Artificial intelligence and convolutional Neural Networks (CNN)to estimate the Body Mass Index (BMI) from human body image using regression models.We get the insparation from estimating bmi from hman photograph [41]

3.2 Methodology

3.2.1 Data collection

Since our project is based on the BMI estimation from the body images we tried to find dataset that contains images of the human body but we only find two small datasets.We tried to find another one but we did not find anything all the available datasets contain face images like the Vip attribute dataset, Face to BMI dataset, Height and weight prediction, all of them don't contain images of the human body so that's why we kept these two datasets which are:

1. The first dataset is celebrity body image download via Python script by recognizing the celebrity's name from their face which contains 451 images. [42]

2. The second dataset is The Photographic Height-Weight Chart via Python script which contains 1821 images. [43]

so the Dataset contains a total of 2,272 images but considered as outliers 444 images so there are 1,828 images available for training and testing. Number of Images: 2272 RGB images.

- **Image Type**: RGB (Red, Green, Blue) images.
- **Data normalization and augmentation**: We use ImageDataGenerator to normalize a labeled and to be augmented by rotating, shifting, and horizontal flipping.

Image	Weight (kg)	Height (cm)
	50	1.52
	104	1.52
	108	1.64
	60	1.67
	77	1.80

Tableau 3.1: The Original Dataset

3.2.2 BMI Calculation

For each record in the dataset, we calculate the Body Mass Index(BMI) using this formula.

$$BMI = \frac{Weight}{(Height)^2} \tag{3.1}$$

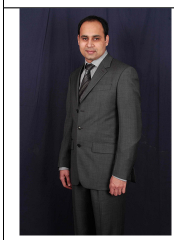
Image	Weight (kg)	Height (m)	BMI
	50	1.52	21.6
	104	1.52	45.01
	108	1.64	40.15
	60	1.67	21.76
	77	1.80	23.76

Tableau 3.2: The Dataset after calculating The BMI

3.2.3 Data segmententation

The dataset is already segmented the was from inspiration from [44]

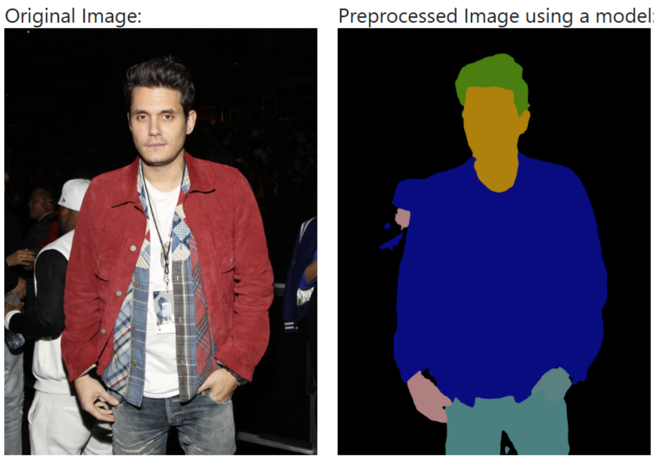


Figure 3.1: The preprocessing image.

And we use a preprocessed images as a filter to get rid of the background by converting all of its color pixels to 1 and multiplying them to the original image.

Figure 3.2: Segmented image.

3.3 Transfer Learning

3.3.1 Reasons for Using Transfer Learning

1. **Limited Data Problem:** The dataset is relatively small (only 2272 samples), and training a deep neural network from scratch can lead to overfitting. Overfitting occurs when the model learns the noise and details in the training data to such an extent that it negatively impacts the model's performance on new data.

2. **Pre-trained Models:** Transfer learning utilizes models pre-trained on large datasets. These models have learned robust features that can generalize well to our dataset. By fine-tuning these models on our specific task, we can leverage these learned features and improve performance even with our limited data.

3.3.2 Pretrained Models Used

1. **ResNet50:** A 50-layer residual network known for its strong performance on image classification tasks.

2. **ResNet101:** A deeper version of ResNet with 101 layers, capable of capturing more complex features.

3. **ResNet152:** An even deeper ResNet with 152 layers, further enhancing the model's ability to learn intricate patterns.

4. **ConvNeXtBase:** A modern architecture designed for high performance and efficiency in image tasks.

5. **MobileNet:** A lightweight model optimized for mobile and embedded vision applications, balancing performance and efficiency.

3.4 Regression model training architecture.

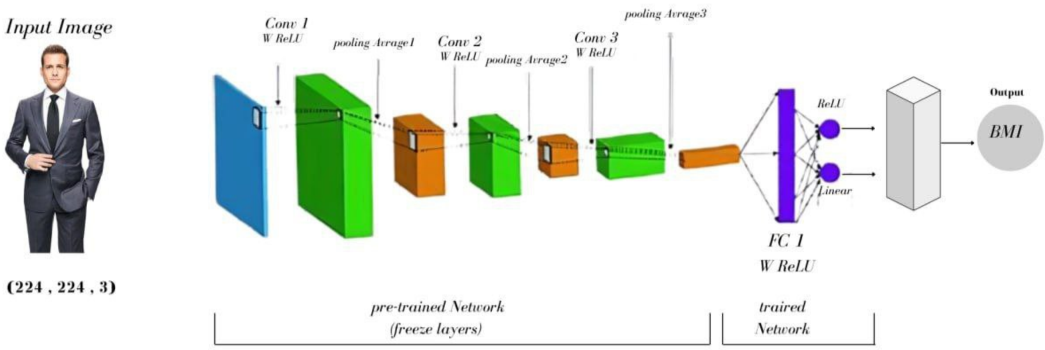


Figure 3.3: Regression model training architecture.

Our transfer learning strategy is designed to maximize the benefits of pre-trained models while adapting them to our specific task of predicting BMI from images. Here's a detailed explanation of our strategy:

3.4.1 Model Freezing and Customization

- **Initial Freezing:** Initially, we froze the layers of the pre-trained model. Freezing layers means that during the initial phase of training, the weights of these layers are not updated. This allows the model to retain the valuable features learned from the large dataset (ImageNet) without disrupting them.

- **Adding Custom Layers:** After the base model, we added custom dense layers tailored for our regression task. These layers include a dense layer with 256 units, a dropout layer to prevent overfitting, and a final dense layer for the BMI output.

Initial Training with Frozen Layers

- **Compile the Model:** We compiled the model with a Stochastic Gradient Descent (SGD) optimizer, Huber loss function, and metrics including Mean Absolute Error (MAE) and Coefficient of Determination (R^2).

- **Train the Model:** We trained the model for a fixed number of epochs (500) with the pre-trained layers frozen. This step allows the custom layers to learn to use the features extracted by the pre-trained layers without altering those features.

3.4.2 Unfreezing and Fine-Tuning

- **Unfreeze Some Layers:** After the initial training phase, we unfroze some or all of the pre-trained layers to fine-tune the entire model. This step allows the entire network to adjust its weights based on our specific dataset, potentially improving performance.

- **Fine-Tune the Model:** Continue training the model for additional epochs with a lower learning rate to fine-tune the weights of the pre-trained layers along with the custom layers. This helps in making fine adjustments that improve model performance.

3.4.3 Learning Rate Scheduling

We used a custom learning rate schedule to adjust the learning rate dynamically during training. This helps in achieving a balance between convergence speed and model performance.

3.4.4 Evaluation and Analysis

After training, the model was evaluated on the test dataset to measure its performance. Metrics such as mean absolute error (MAE) and coefficient of determination (R^2) were recorded.

Summary: Our training strategy in transfer learning involves a systematic approach of initially leveraging pre-trained models by freezing their layers, then gradually unfreezing and fine-tuning these layers to adapt to our specific task. This strategy allows us to utilize the robust feature extraction capabilities of pre-trained models while adapting them to predict BMI from images effectively. By combining techniques like layer freezing, custom layers, learning rate scheduling, and thorough evaluation, we ensure the model achieves high performance on our dataset.

3.5 Training and Evaluation

1. **Training:** Each model is trained for 500 epochs on the training data.

2. **Validation:** Performance is evaluated on the validation set, tracking validation loss and metrics.

3. **Repetition:** The process is repeated five times for each model to ensure robustness and reliability of the results.

3.5.1 Pre-trained Base Model: ResNet50

Pretrained Model	Metrics	Train	Test
ResNet50	Coefficient of Determination (R^2)	36.3%	35.7%
	Loss	3.5012	3.4878
	Mean Absolute Error	3.9704	3.9495

Tableau 3.3: Regression module using pretrained model (ResNet50) training and testing results.

3.5.2 Pre-trained Base Model: ResNet101

Pretrained Model	Metrics	Train	Test
ResNet101	Coefficient of Determination (R^2)	42.7%	37.4%
	Loss	3.3482	3.6792
	Mean Absolute Error	3.8152	4.1506

Tableau 3.4: Regression module using pretrained model (ResNet101) training and testing results.

3.5.3 Pre-trained Base Model: ResNet152

Pretrained Model	Metrics	Train	Test
ResNet152	Coefficient of Determination (R^2)	42.5%	30.3%
	Loss	3.328	3.763
	Mean Absolute Error	3.79	4.23

Tableau 3.5: Regression module using pretrained model (ResNet152) training and testing results.

3.5.4 Pre-trained Base Model: Convnext Base

Pretraind Model	Metrices	Train	Test
ConvNext Base	Coefficient of Determination (R^2)	65.5%	43.2%
	Loss	2.45	3.19
	Mean-absolute error	2.9	3.65

Tableau 3.6: Regression module using pretrained model(ConvNext Base) training and testing results

3.5.5 Pre-trained Base Model: MobileNet

Pretraind Model	Metrices	Train	Test
MobilNet	Coefficient of Determination (R^2)	28.4%	28.2%
	Loss	3.70	3.9
	Mean-absolute error	4.17	4.42

Tableau 3.7: Regression module using pretrained model(Mobile Net) training and testing results

From these results we can see that the our regression model has the best results using convNextBase,this is why we put our focus on to make it better by changing **Hyperparameter Tuning and Optimization Learning Rate and Optimization Strategy** and that happened by utilizing a custom learning rate scheduler, which adjusts the learning rate dynamically during training depending on the number of eppochs. This helps in better convergence and prevents overshooting or getting stuck in local minima. also, The optimizer used (SGD with momentum) is effective for training, provided the learning rate schedule is appropriately tuned.

Pretraind Model	Metrices	Train	Test
ConvNext Base	Coefficient of Determination (R^2)	87.6%	77.2%
	Loss	1.308	1.97
	Mean-absolute error	1.74	2.4

Tableau 3.8: The enhanced regression module using the pretrained module (ConvNextBase results

3.6 The original model Vs. Our model

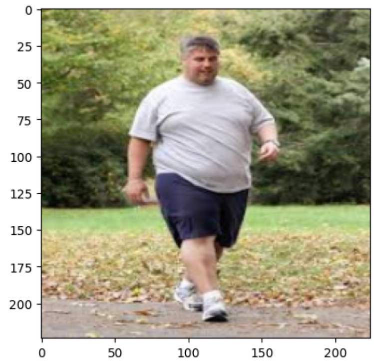
We can compare our model and the original in this table :

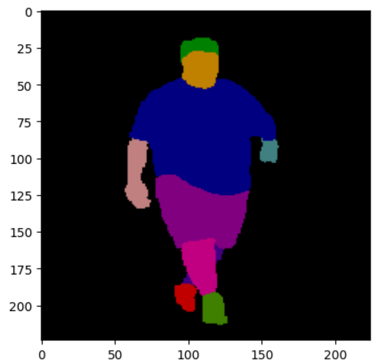
Metric	**Original Model**	**Our Model**
Coefficient of Determination (Train)	0.407	0.8776
Coefficient of Determination (Test)	0.236	0.7725
Loss (Train)	3.142	1.9738
Loss (Test)	4.191	1.9738
Mean Absolute Error (Train)	3.605	1.7431
Mean Absolute Error (Test)	4.672	2.4258

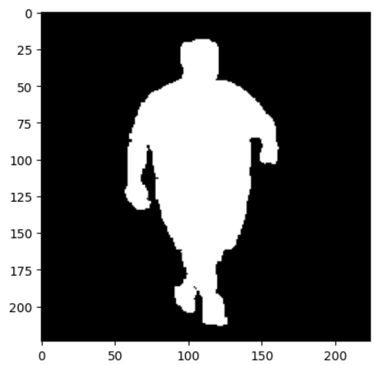
Tableau 3.9: Comparison between the Original Model and Our Model Results

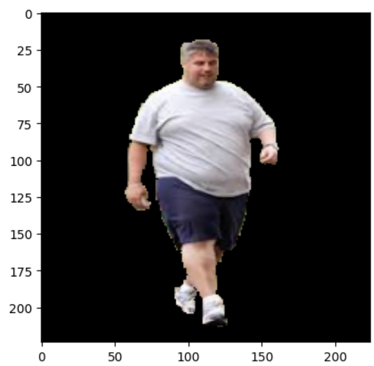
3.7 Test with image (Our model)

After choosing the image this will show:









1. if bmi pred < 15:**Very severely underweight**
2. elif 15 <= bmi pred < 16:**Severely underweight**
3. elif 16 <= bmi pred < 18.5:**Underweight**
4. elif 18.5 <= bmi pred < 25:**Normal**
5. elif 25 <= bmi pred < 30:**Overweight**
6. elif 30 <= bmi pred < 35:**Moderately obese**
7. elif 35 <= bmi pred < 40: **Severely obese**
8. elif bmi pred >= 40: **severely obese**

Output:

1/1 [==============================] - 0s 73ms/step
1/1 [==============================] - 0s 67ms/step
BMI: 27.43268585205078
Overweight

3.8 Classification Model:

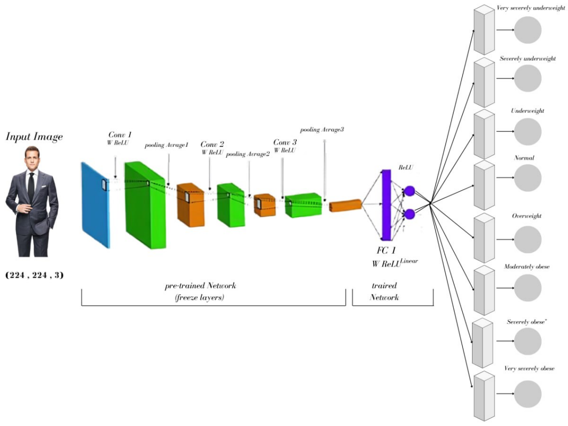


Figure 3.4: Classification Model.

We choose 8 classes to estimate the body mass index from which are :
0 : Very severely underweight ,
1 : Severely underweight ,
2 : Underweight,
3 : Normal ,
4 : Overweight,
5 : Moderately obese ,
6 : Severely obese,
7 : Very severely obese

3.9 Classification model results

Metric	Training	Validation
Loss	0.8234	1.4865
Sparse Categorical Accuracy	0.6728	0.4590

Tableau 3.10: Model Training Results

3.10 Futur work

Our project is based on human weight estimation from body images. This approach is very important in especially in our world where technology plays a crucial role in our daily lives so this approach must spread especially in health. The futur advancement of this approach in our opinion is to estimate the BMI not only for one person in the image but for many persons in the same time. which can save time and also better performance.

3.11 Conclusion

In this chapter, we talk about our dataset and the reason behind choosing transfer learning in our model we also explore our regression model architecture input-output and the different prtrianed models that we use, and then we mention the reason why we got better results than the original model.

Chapter 4
Supplementry material

Chapter 4

Supplementry material

4.1 Introduction

In this chapter, we will go over the set of programming languages, libraries, and development tools that we used during the development process of our project. These tools and libraries are considered essential in building and developing modern software, as they help improve efficiency, simplify operations, and provide integrated solutions to technical problems that we may encounter. In this chapter, we will review these tools, as follows

4.2 Development Tools

4.3 Artificial Intelligence

Artificial Intelligence (AI) refers to the capability of computer systems to perform tasks that typically require human intelligence. These tasks include reasoning, learning, problem-solving, perception, and language understanding. AI can be implemented in various forms, from simple rule-based systems to complex neural networks and machine learning models. [?]

4.4 Convolutional Neural Network (CNN)

A convolutional neural network (CNN) is a conditioned type of feedforward neural network that uses filter (or kernel) optimization to learn feature evolution. By using regularized weights over fewer connections, the vanishing and exploding gradients observed with backpropagation in previous neural networks can be prevented [45, 46] for example, each neuron in a fully connected layer requires 10,000 weights to process an image of size

100 100 pixels. However, when using cascaded convolution kernels (or cross-correlation kernels) [47,48], only 25 neurons are needed to process tiles of size 5x5 [49,50]. Higher-level features are extracted from a wider context window than lower-level features

4.5 Machine Learning(ML)

Machine learning, an integral part of artificial intelligence (AI), empowers machines to learn from data and accomplish tasks that typically necessitate human intelligence, By utilizing algorithms. machine learning allows machines to access, analyze, and discern patterns in data, subsequently constructing models capable of making predictions or decisions without explicit programming, The applications of machine learning are diverse, encompassing image recognition, and data analysis.

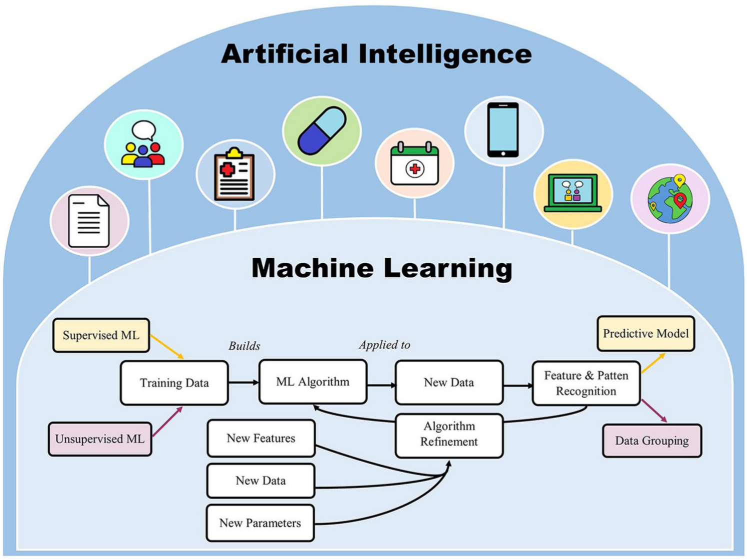


Figure 4.1: Understanding AI and ML Interconnectivity.

4.6 Transfer learning

Transfer learning is a machine learning technique that involves reusing a model developed for a particular task as the starting point for a model on a second task. It is a deep learning technique that enables developers to harness a neural network used for one task and apply it to another domain [?].

4.6.1 The Used Programming Language (Python)

Python is a high-level, interpreted, and object-oriented programming language known for its dynamic semantics and easy-to-learn syntax. It supports rapid application development and can be used as a scripting language or for connecting existing components. Python's productivity benefits include fast edit-test-debug cycles and easy debugging due to its exception handling. It also offers a source-level debugger for detailed inspection of variables and code execution [?].

4.6.2 Software and frameworks

4.6.3 Jupyter Notebook

The term "notebook" refers to a variety of objects, including Jupyter documents, Jupyter web applications, and the Jupyter Python web server. These notebooks are composed of code units with the option to insert text or data to explain how to execute them. Each cell is contained in a document that can be easily shared with other users. The tool has a wide range of applications, including statistical modeling, numerical simulations, data translation, and even machine learning [?]

4.6.4 Google Colab

Google Colab or Collaboratory is a free Jupyter Notebook-based cloud service for machine learning training and research12. It is offered by Google Research where users can write and execute source code in an editor and then run it through a browser.

4.6.5 Used libraries

4.6.5.1 Open CV:

OpenCV, an acronym for Open-Source Computer Vision Library, is widely recognized as the go-to solution for computer vision applications. With its comprehensive documentation and user-friendly infrastructure, OpenCV seamlessly integrates numerous cutting-edge computer vision algorithms.

It boasts compatibility with multiple programming languages, such as Python, C++, and Java, making it a versatile tool for developers. From identifying objects and faces to recognizing handwriting, OpenCV empowers users to process images and videos with exceptional precision and efficiency [?].

4.6.5.2 NumPy:

NumPy is a Python library for scientific computing that offers a multidimensional array object, derived objects, and a wide range of functions for efficient array operations.

It supports mathematical operations, array manipulation, sorting, I/O operations, linear algebra, statistical calculations, random simulation, and more [?]

4.6.5.3 pandas (pd):

The open-source data analysis and manipulation program Pandas is quick, strong, adaptable, and simple to use. It offers a rich range of functions for working with structured data and is built on the Python programming language [?]

4.6.5.4 matplotlib.pyplot (plt):

matplotlib.A tool called pyplot offers a MATLAB-like interface for matplotlib plotting. It has several methods that make it easier to create plots.

In addition to opening figures on the screen and acting as a GUI manager for figures, it enables interactive plot development. For simple programmatic plot creation and interactive charts, pyplot is extremely helpful [?]

4.6.5.5 Keras:

Keras is a Python library that sits on top of the TensorFlow platform, serving as a high-level API. It simplifies creating and training deep learning models, emphasizing fast debugging, elegant and concise code, easy maintenance, and efficient deployment.

Keras offers crucial abstractions and foundational components for developing and delivering machine learning solutions with rapid iteration velocity [?]

4.6.6 Time:

Python's "**time**" library enables developers to effortlessly manipulate dates and times within their programs. This comprehensive library offers a wide range of functions designed to facilitate time measurement operations, including code execution time and date/time manipulation [?].

4.6.6.1 Gdown:

Gdown, a Python library, streamlines the process of downloading public files and folders from Google Drive. Its advanced features surpass the capabilities of conventional tools such as curl or widget, offering an enhanced experience for Google Drive downloads [?].

4.6.6.2 Pathlib:

Pathlib is a powerful Python library for handling file system paths in an object-oriented manner it provides classes that represent paths with semantics appropriate for different operating systems [?].

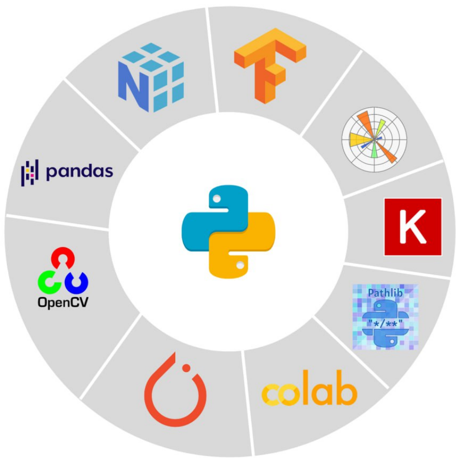


Figure 4.2: Used libraries.

4.7 Workstation

Nom de l'appareil	DESKTOP-RDC9CE4
Processeur	Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz 1.99 GHz
Mémoire RAM installée	16,0 Go (15,8 Go utilisable)
Type du système	Système d'exploitation 64 bits, processeur x64
Stylet et fonction tactile	La fonctionnalité d'entrée tactile ou avec un stylet n'est pas disponible

Tableau 4.1: Informations sur l'ordinateur

4.8 Code documentation

4.8.0.1 Data Processing Code:

Our code reads data from two Csv files:"height-weight-chartdataset.csv"and"celebdatasets.csv". It combines the data from these files into a single data frame called df.

```
import pandas as pd

hwcd_df = pandas.read_csv("/content/BMI-prediction-from-Human-Photograph-main/height-weight-chart_dataset.csv")
celeb_df = pandas.read_csv("/content/BMI-prediction-from-Human-Photograph-main/celeb_datasets.csv")

df = pd.concat( [hwcd_df, celeb_df])
```

Figure 4.3: Merging datasets in Python.

4.9 Model Creation Code

The Keras library is used to create a neural network model based on the ResNet101 architecture, which is one of the deep neural networks designed for image processing.

```
base_model = keras.applications.ResNet101(
    include_top=False,
    weights="imagenet",
    input_shape=input_shape,
    pooling="avg"
)

base_model.trainable = False
base_model.summary()
```

Figure 4.4: This code initializes a ResNet101 model without its top layer, using ImageNet weights, and specifies input shape and pooling. Then, it freezes the model's weights to prevent further training.

4.10 Model training code

A neural network model is created and trained using the Keras library over several different experiments, and the trained model is then saved for each experiment. The goal of this process is to evaluate the model's performance across multiple experiments and improve it based on the results. In this code, an empty list called 'VALIDATION LOSS' is created to store the validation loss values for each experiment. The process of creating and training the model is repeated five times, each time creating a new model using 'Sequential'.

In each experiment, a dense 'Dense' layer with 256 units and a ReLU activation function (with 2048 inputs) is added to the model, followed by a 'Dropout' layer with a drop rate of 25% to avoid overfitting, and finally a final 'Dense' layer with 1 unit and a linear activation function. .

The model is assembled using the SGD optimizer, and trained using the training and validation data specified for each experiment. Finally, the trained model is saved in a file with a name that reflects the experiment number. The model is then assembled using the SGD optimizer with a learning rate of 0.00001 and momentum of 0.9, with a Huber loss function and the accuracy metrics mean absolute error and coeff determination. The model is trained using training and validation data for each trial for 500 epochs, with the training progress displayed. Two calls are used during training: PlotLossesKeras to keep track of the loss and LearningRateScheduler to update the learning rate.

```
VALIDAITON_LOSS = []

for j in range(5):
  model = Sequential()
  model.add(layers.Dense(256, activation='relu', input_dim=2048))
  model.add(layers.Dropout(0.25))
  model.add(layers.Dense(1, activation='linear'))

  model.compile(optimizer=keras.optimizers.SGD(
      learning_rate=1e-5,
      momentum=0.9
  ), loss=keras.losses.Huber(), metrics=["mean_absolute_error", coeff_determination])

  history = model.fit(train_features[j], train_labels[j], epochs=500,
    validation_data=(test_features[j], test_labels[j]),
    verbose=2,
    callbacks=[
        PlotLossesKeras(),
        keras.callbacks.LearningRateScheduler(lr_scheduler),
    ]
  )
  model.save(f"last_model{j}.h5")

  results = model.evaluate(test_features[j], test_labels[j])
  VALIDAITON_LOSS.append(results)
```

Figure 4.5: This code iteratively trains and evaluates a neural network model for regression five times, storing the validation loss for each iteration in a list named ‘VALIDATION LOSS‘.

4.11 Model evaluation code

The ‘VALIDATION LOSS‘ function was used to evaluate the model’s performance across five different experiments. The list contains the validation loss results for each experiment, where each result includes three values: the validation loss value, the mean absolute error, and the coefficient of determination. The results show the following values:

```
VALIDAITON_LOSS
```

```
[[3.767411708831787, 4.2398457527160645, 0.21404393017292023],
 [4.1873955726623535, 4.661474227905273, 0.23661932349205017],
 [3.7670698165893555, 4.234373092651367, 0.26941707730293274],
 [3.7532243728637695, 4.224757671356201, 0.2328120321035385],
 [3.6791768074035645, 4.150638103485107, 0.37444713711738586]]
```

Figure 4.6: Validation loss metrics for a regression model.

General Conclusion

Our thesis addresses the topic of computer vision-based BMI weight estimation. Our project seeks to BMI estimation from human body images. In our project process, we faced multiple challenges: at the beginning, we estimated animal weight but since the estimation of animal weight requires specialized cameras:3D cameras, RGB cameras, and lidar devices. which is costly also these cameras are not availaible. That's why we switched to human weight estimation. We also faced challenges in this approach here are:

1. **Dataset availability:** Since our project is based on BMI recognition from human body images we tried to find a suitable that large but we could not find a large dataset at all. But we find two small datasets which are small and combine them together

2. **Training challenges:**In training the model both of our computers take a lot of time while training also the training never stops like it's an infinite loop that's why we Borrowed another computer

3. **Google colab time** Google Colab was the best solution for running our code it's fast and easy but it gives us just 2 hours per day. Due to that, we create so many emails so we can run our code more easily.

The implementation of our model was good using the pretrained model convNextBase Coefficient of Determination for training is:87% and 77,5% for test. The loss and mean absolute error metrics showed a significant improvement compared to the original model. The successful application of this transfer learning approach not only illustrates the efficacy of pre-trained models in small data scenarios but also highlights the potential for broader adoption of AI-driven techniques in health and wellness applications. Moving forward, future work could focus on refining the model for multi-person BMI estimation in a single image, enhancing its utility in healthcare scenarios where simultaneous analysis of multiple individuals is beneficial.

As students, our engagement in this project has granted us invaluable benefits. We have gained hands-on experience to understand more about Python get more experience in it, also we learn more about Canva to design specific images especially in the second chapter.

Bibliography

[1] Yuliang Zhao, Qijun Xiao, Jinhao Li, Kaixuan Tian, Le Yang, Peng Shan, Xiaoyong Lv, Lianjiang Li, and Zhikun Zhan. Review on image-based animals weight weighing. *Computers and Electronics in Agriculture*, 215:108456, 2023.

[2] Christian Pfitzner, Stefan May, and Andreas Nüchter. Body weight estimation for dose-finding and health monitoring of lying, standing and walking patients based on rgb-d data. *Sensors*, 18(5):1311, 2018.

[3] Renê Ripardo Calixto, Luis Gonzaga Pinheiro Neto, Tarique da Silveira Cavalcante, Márcio Facundo Aragão, and Ebenezer de Oliveira Silva. A computer vision model development for size and weight estimation of yellow melon in the brazilian northeast. *Scientia Horticulturae*, 256:108521, 2019.

[4] Maria Q Feng, Ryan Y Leung, and Casey M Eckersley. Non-contact vehicle weigh-in-motion using computer vision. *Measurement*, 153:107415, 2020.

[5] Jeanne Oui. L'agriculture rêve-t-elle de moutons électriques? paradoxes des technologies numériques pour la transition environnementale de l'agriculture. *Multitudes*, (3):66–72, 2023.

[6] Yuliang Zhao, Qijun Xiao, Jinhao Li, Kaixuan Tian, Le Yang, Peng Shan, Xiaoyong Lv, Lianjiang Li, and Zhikun Zhan. Review on image-based animals weight weighing. *Computers and Electronics in Agriculture*, 215:108456, 2023.

[7] Weihong Ma, Xiangyu Qi, Yi Sun, Ronghua Gao, Luyu Ding, Rong Wang, Cheng Peng, Jun Zhang, Jianwei Wu, Zhankang Xu, et al. Computer vision-based measurement techniques for livestock body dimension and weight: A review. *Agriculture*, 14(2):306, 2024.

[8] Hera Siddiqui, Ajita Rattani, Karl Ricanek, and Twyla Hill. An examination of bias of facial analysis based bmi prediction models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2926–2935, 2022.

[9] Abozar Nasirahmadi, Sandra A Edwards, and Barbara Sturm. Implementation of machine vision for detecting behaviour of cattle and pigs. *Livestock Science*, 202:25–38, 2017.

[10] Jørgen Kongsro. Estimation of pig weight using a microsoft kinect prototype imaging system. *Computers and Electronics in Agriculture*, 109:32–35, 2014.

[11] Mohammadamin Kashiha, Claudia Bahr, Sanne Ott, Christel PH Moons, Theo A Niewold, Frank O Ödberg, and Daniel Berckmans. Automatic weight estimation of individual pigs using image analysis. *Computers and Electronics in Agriculture*, 107:38–44, 2014.

[12] Arthur FA Fernandes, João RR Dórea, Robert Fitzgerald, William Herring, and Guilherme JM Rosa. A novel automated system to acquire biometric and morphological measurements and predict body weight of pigs via 3d computer vision. *Journal of animal science*, 97(1):496–508, 2019.

[13] Alexey Ruchay, Vitaly Kober, Konstantin Dorofeev, Vladimir Kolpakov, and Sergei Miroshnikov. Accurate body measurement of live cattle using three depth cameras and non-rigid 3-d shape recovery. *Computers and Electronics in Agriculture*, 179:105821, 2020.

[14] LI Jiawei, MA Weihong, LI Qifeng, XUE Xianglong, and WANG Zhiquan. Automatic acquisition and target extraction of beef cattle 3d point cloud from complex environment. *Smart Agriculture*, 4(2):64, 2022.

[15] Hurriyatul Fitriyah and Gembong Edhi Setyawan. Automatic estimation of human weight from body silhouette using multiple linear regression. In *2018 5th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, pages 749–752. IEEE, 2018.

[16] Tan Xiaohui, Peng Xiaoyu, Liu Liwen, and Xia Qing. Automatic human body feature extraction and personal size measurement. *Journal of Visual Languages & Computing*, 47:9–18, 2018.

[17] Benjamin M Auerbach and Christopher B Ruff. Human body mass estimation: a comparison of "morphometric" and "mechanical" methods. *American Journal of Physical Anthropology: the Official Publication of the American Association of Physical Anthropologists*, 125(4):331–342, 2014.

[18] Xingguo Liu, Jianwei Niu, Linghua Ran, and Taijie Liu. Estimation of human body volume (bv) from anthropometric measurements based on three-dimensional (3d) scan technique. *Aesthetic Plastic Surgery*, 41:971–978, 2017.

[19] Christian Pfitzner. *Visual Human Body Weight Estimation with Focus on Clinical Applications*. Universität Würzburg, 2019.

[20] Ilaha Tagiyeva. Ensuring global food safety. *Economic and Social Development: Book of Proceedings*, 3:29–35, 2020.

[21] Kaiqian Peng, Wei Ma, Jinzhu Lu, Zhiwei Tian, and Zhiming Yang. Application of machine vision technology in citrus production. *Applied Sciences*, 13(16):9334, 2023.

[22] Javier Tardáguila, Maria P Diago, B Millan, José Blasco, Sergio Cubero, and Nuria Aleixos. Applications of computer vision techniques in viticulture to assess canopy features, cluster morphology and berry size. In *I International Workshop on Vineyard Mechanization and Grape and Wine Quality 978*, pages 77–84, 2020.

[23] Weilin Wang and Changying Li. Size estimation of sweet onions using consumer-grade rgb-depth sensor. *Journal of food engineering*, 142:153–162, 2014.

[24] Weilin Wang and Changying Li. A multimodal machine vision system for quality inspection of onions. *Journal of food engineering*, 166:291–301, 2015.

[25] M Omid, M Khojastehnazhand, and A Tabatabaeefar. Estimating volume and mass of citrus fruits by image processing technique. *Journal of food Engineering*, 100(2):315–321, 2010.

[26] Donggil Lee, Seonghoon Kim, Miseon Park, and Yongsu Yang. Weight estimation of the sea cucumber (stichopus japonicas) using vision-based volume measurement. *Journal of Electrical Engineering and Technology*, 9(6):2154–2161, 2014.

[27] Somaye Amraei, Saeed Ali Mehdizadeh, Mohammad Reza Alizadeh, Hossein Rahimizadeh, Leila Naderloo, and Mohammad Javad Kasaei. Development of a transfer function for weight prediction of live broiler chicken using machine vision. *Measurement*, 180:109597, 2021.

[28] Fang Zhao, Hongnian Yu, Min Fan, Shouling Ji, Zhi Wang, and Dazhao Chen. Broiler weight estimation based on convolutional neural network and principal component regression using real and synthetic depth images. *Computers and Electronics in Agriculture*, 181:105926, 2021.

[29] Jim R Parker. *Algorithms for image processing and computer vision*. John Wiley & Sons, 2016.

[30] Xiaoyu Li, Bo Zhang, Pedro V Sander, and Jing Liao. Blind geometric distortion correction on images through deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4855–4864, 2019.

[31] Weihong Ma, Yi Sun, Xiangyu Qi, Xianglong Xue, Kaixuan Chang, Zhankang Xu, Mingyu Li, Rong Wang, Rui Meng, and Qifeng Li. Computer-vision-based sensing technologies for livestock body dimension measurement: A survey. *Sensors*, 24(5):1504, 2024.

[32] Enes Kocabey, Mustafa Camurcu, Ferda Ofli, Yusuf Aytar, Javier Marin, Antonio Torralba, and Ingmar Weber. Face-to-bmi: Using computer vision to infer body mass index on social media. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11, pages 572–575, 2017.

[33] Rohit Sharma, Mahesh Kumar, and MS Alam. Image processing techniques to estimate weight and morphological parameters for selected wheat refractions. *Scientific Reports*, 11(1):20953, 2021.

[34] Nadeem Yousaf, Sarfaraz Hussein, and Waqas Sultani. Estimation of bmi from facial images using semantic segmentation based region-aware pooling. *Computers in Biology and Medicine*, 133:104392, 2021.

[35] Rohan Soneja, S Prashanth, and R Aarthi. Body weight estimation using 2d body image. *International Journal of Advanced Computer Science and Applications*, 12(4), 2021.

[36] Baohua Zhang, Ning Guo, Jichao Huang, Baoxing Gu, and Jun Zhou. Computer vision estimation of the volume and weight of apples by using 3d reconstruction and noncontact measuring methods. *Journal of Sensors*, 2020:1–12, 2020.

[37] Wolfgang Förstner and Bernhard P Wrobel. *Photogrammetric computer vision*. Springer, 2016.

[38] AJ Heinrichs, GW Rogers, and JB Cooper. Predicting body weight and wither height in holstein heifers using body measurements. *Journal of dairy science*, 75(12):3576–3581, 2014.

[39] Carsten Enevoldsen and Troels Kristensen. Estimation of body weight from body size measurements and body condition scores in dairy cows. *Journal of dairy science*, 80(9):1988–1995, 1997.

[40] LA González, G Bishop-Hurley, D Henry, and E Charmley. Wireless sensor networks to study, monitor and manage cattle in grazing systems. *Animal Production Science*, 54(10):1687–1693, 2014.

[41] atoms18. Bmi-prediction-from-human-photograph, 2015.

[42] height-weight-chart author=atoms18, year=2015, url=www.height-weight-chart.com/heightweight.html .

[43] Antitza Dantcheva, Francois Bremond, and Piotr Bilinski. Show me your face and i will tell you your height, weight and body mass index. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 3555–3560. IEEE, 2018.

[44] Peike Li, Yunqiu Xu, Yunchao Wei, and Yi Yang. Self-correction for human parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(6):3260–3271, 2020.

[45] Ragav Venkatesan and Baoxin Li. *Convolutional neural networks in visual computing: a concise guide*. CRC Press, 2017.

[46] Valentina E Balas, Raghvendra Kumar, Rajshree Srivastava, et al. *Recent trends and advances in artificial intelligence and internet of things*. Springer, 2020.

[47] Yingjie Zhang, Hong Geok Soon, Dongsen Ye, Jerry Ying Hsi Fuh, and Kunpeng Zhu. Powder-bed fusion process monitoring by machine vision with hybrid convolutional neural networks. *IEEE Transactions on Industrial Informatics*, 16(9):5769–5779, 2019.

[48] Nikolay I Chervyakov, Pavel Alekseevich Lyakhov, Maxim Anatolievich Deryabin, NN Nagornov, Maria V Valueva, and Georgii V Valuev. Residue number system-based solution for reducing the hardware cost of a convolutional neural network. *Neurocomputing*, 407:439–453, 2020.

[49] Hamed Habibi Aghdam, Elnaz Jahani Heravi, et al. Guide to convolutional neural networks. *New York, NY: Springer*, 10(978-973):51, 2017.

[50] Les Atlas, Toshiteru Homma, and Robert Marks. An artificial neural network for spatio-temporal bipolar patterns: Application to phoneme classification. In *Neural Information Processing Systems*, 2018.