Age & Gender Recognition In The Wild

Bachelor's Final Thesis

Degree in Data Science & Engineering

Emili Bonet i Cervera
emili.bonet@estudiantat.upc.edu

Tutored and supervised by

Javier Ruiz-Hidalgo, Josep Ramon Morros Rubió, and Mar Ferrer Ferrer

June 15, 2022

Abstract

The estimation of demographics from image data, and in particular of gender and age, is a subject with an extensive amount of applications. However, current state-of-the-art is not entirely focused on realistic and unconstrained scenarios, which makes those approaches unusable for certain real-life settings. This thesis analyzes the issue of robust age and gender prediction, and proposes a new paradigm to build upon an alternative framework from which methods that are more capable in realistic situations can be developed. Namely, we present a method based on Deep Neural Networks (DNNs) that acts as an ensemble model, including predictions from both corporal and facial features. Thus, our model can act both when faces are not very visible or are occluded, and can take advantage of the extra information when they are visible. The system presented combines multiple off-the-shelf models such as RetinaFace and ShuffleNet for facial tasks, and Faster R-CNN with ResNet backbone pre-trained on COCO for human detection. From my side, a module was trained to predict gender and age based on body detections, where EfficientNet is used as backbone. Consequently, it was demonstrated that body-based models have the capacity to be more resilient.

Resum

L'estimació de grups demogràfics a partir d'imatges, i en particular pel que fa a l'estimació d'edat i sexe, és un sector amb un ampli ventall d'aplicacions. Tanmateix, l'estat de l'art actual està poc encarat a escenaris realistes que no contemplen cap mena de restriccions, la qual cosa fa que els seus mètodes siguin inservibles per certs tipus de dades de la vida real. Aquesta tesi analitza la qüestió de la predicció robusta per sexe i edat, i proposa un nou paradigma per construir un marc de treball alternatiu des d'on desenvolupar mètodes més capaços d'actuar en situacions realistes. En concret, es demostra empíricament com l'estat de l'art basat en trets facials no és capaç d'actuar al nostre conjunt de dades que representen aquestes situacions realistes, i presentem un mètode basat en Xarxes Neuronals Profundes (DNNs, per la seva abreviació en anglès) que actua com un model de predicció conjunta, incloent-hi prediccions fetes a partir de característiques extretes de tot el cos a més a més de les aconseguides a través del rostre. Això permet al model actuar quan les cares són poc visibles o estan obstruïdes, i aprofitar-se de la informació addicional quan aquestes són visibles. El sistema presentat combina diversos models aplicats en fred, com per exemple RetinaFace i ShuffleNet per tasques facials, i una Faster R-CNN pre-entrenada amb COCO amb una ResNet com a model vertebral per detecció humana. Per la meva part, també s'ha entrenat un mòdul per predir sexe i edat a partir de deteccions corporals, on es fa servir EfficientNet com a vertebral. Consegüentment, s'ha demostrat que els models basats en cos tenen la capacitat de ser més resilients.

Keywords: Computer Vision, Deep Learning, In-the-wild, Human detection, Face detection, Age recognition, Gender recognition.

Acknowledgements

My wholehearted gratitude goes to my supervisors, specially to Mar Ferrer for meeting with me countless times in order to turn this endeavour into such a fruitful experience, and also to Javier Ruiz and Ramon Morros, who helped me through this journey by providing much needed knowledge and expertise.

I am immensely grateful to all the colleagues that selflessly aided me, and in particular to Carlos Hernandez for his invaluable assistance with Weights & Biases. Many thanks also to all my friends were so keen on giving constructive criticism, thus helping me refine my project into the masterpiece it is now.

Finally, I would also like to mention my family, as I always had their unconditional support to motivate and inspire me to push through my work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Artificial Intelligence (AI) has played an important role in the latest technological innovations in many fields, and is being used by businesses in many sectors. Currently, AI has achieved a remarked presence in the products and pipelines of leading companies who seek to improve their productivity and competency over rival brands. Considering the vast amount of content that these companies produce on a daily basis, AI has become a key component in the path towards the automatization of most processes. The application of AI-related techniques can be seen in a wide variety of sectors: from healthcare with systems able to automatize triage, to business and economics for predicting trends and designing policies, passing through agriculture to detect diseases in plants, and even in education with personalized learning processes; AI has come to revolutionize our societies in every sense.

Our project will involve the particular case of Computer Vision, which is one of the most frequent AI techniques employed in the industries. The recent developments in Deep Learning in Computer Vision over the past couple of decades have produced techniques that managed to set new state-of-the-art standards in the field. Computer Vision has seen a tremendous improvement over the now outdated classical approach to vision, and this has provided many tools based on Neural Networks with which one can design and train automated systems that can work with image-based data with remarkable accuracy, easily managing to overcome performance of pre-Deep Learning methods. Despite all of this, there is room for some fair criticism to these new approaches. Mainly, this increase in performance comes at the cost of requiring prodigious amounts of training data and computational resources to train the models and to use them for inference. Furthermore, the lack of interpretability of such models is also a typical point of contention in circumstances where knowing why a certain decision is taken is arguably as important as the decision itself.

Perhaps the best example of Computer Vision being used in a real-world setting would be the renowned autonomous driving, although other more "mundane" tasks such as the detection of cancerous tumors based on medical imagery, systems for crop grading and sorting, or performance assessment in sports have also managed to leverage Computer Vision to obtain excellent results.

1.1 The Application

As the main introductory part of this project has stated, we will employ Computer Vision to solve the requirements of Viume, which is an enterprise that offers Software as a Service products, and is self-described as a company *"Enabling personalized interactions with each user through AI"*. As a matter of fact, this thesis is being developed during an internship inside this company, where the research and development done in it constitutes the foundations of this work.

Viume has been hired by other companies to develop different projects. Its main area of expertise resides in the fashion industry, helping e-commerces to create more scalable pipelines for their sales. The tools used for such purposes are automated product tagging and visual search. My project is focused towards the latter, which consists on looking for similar items for each item of clothing that an e-commerce has.

Visual searchers can greatly benefit from having demographic information regarding the gender and approximate age of the clothes' target demographic. With it, conditions to the search space for the clothes themselves can be applied, and thus better recommendations can be formulated in order to narrow down the search in the database to a reduced dataset, since they avoid major inconsistencies in the final results. However, traditional approaches can face some difficulties when being applied in such contexts, as they typically solely rely on facial attributes. Indeed, models in fashion magazines may appear with their faces turned away from the camera, have been heavily retouched with cosmetics and graphics editing, or simply have their faces cut off from the image, all of which can be rather troublesome for conventional methods.

Viume also has several projects outside the fashion market which are more related to the in-site consumer experience (inside a museum, a supermarket, etc.). They consist on finding the behavioural patterns that a sub-group of people might have. For example, children might always stop in front of the candy aisle, whereas adults do not.

Just as before with the fashion visual searcher, demographics play an important role when it comes to a supermarket or museum's ability to engage their target audience. As with any business open to the public, they both have a keen interest in studying how the public interacts with their products. In order to have more information about the visitors, it was proposed that CCTV cameras that are usually already installed can be used to identify the gender and age of the visitors, and from here analyze the relation of each demographic group with the expositions or products in display. Indeed, in order to elaborate these patterns, each individual must be tagged, and some of the most popular tags to use are gender and age. Going beyond, it can also be interesting to try sentiment analysis to study the reaction to a certain exposition or product, but this is outside the scope of this project. However, an important technical detail is worth considering here: similarly to the fashion visual searcher, performing age and gender recognition in this context might also prove to be difficult if face-based methods were to be applied, as CCTVs usually have low quality images and people normally stand relatively far from the camera. As a result, the faces might appear to have a very low resolution, and therefore facial attributes may not be usable.

In summary, despite the active involvement of Computer Vision in the industry, it became clear that there is a pressing need for a robust age and gender recognition approach suitable for in-the-wild contexts. Even though facial traits contain much information regarding both attributes, it is not wise to rely entirely on this source of information. Therefore, this project is set to explore the possibility of inferring gender and age based on learned body features in addition to facial features. In doing so, I hope to achieve a level of robustness superior to most state of the art methods.

1.2 Objectives and Scope

This project was carried out through a student internship in Viume, a software company that provides AI-driven solutions to e-commerce platforms, in conjunction with UPC's Image Processing Group from the Signal Theory and Communications Department. The purpose of this collaboration was to perform R+D activities regarding the task of gender and age recognition based on visual data. Consequently, I expected to deliver a proof of concept of an alternative implementation to the current product, which resulted in the integration of our new method into Viume's software. In accordance to the aforementioned setting for this project, the following objectives were defined:

1. Perform comprehensive research in the state of the art technology for human detection (full-body and face detection) and gender/age recognition.

2. Look for external datasets and optimal methods for data augmentation for these tasks.

3. Design, implement and train an architecture able to solve the problem of gender and age recognition on the image dataset provided by Viume.

In achieving all these three objectives, I expect to have proven that the approach I introduce is considerably more robust than other state-of-the-art methods, and is thus better suited to perform in-the-wild than its counterparts.

1.3 Project Planning

To guide the project's development towards those goals, the following project planning was designed, consisting of three main stages:

1. *Preparation stage*

   The Project Proposal and Work Plan document was written and delivered, I acquired Viume's dataset to familiarize myself with it and performed descriptive analysis as well as cleaning and formatting to suit subsequent development, whenever necessary, and where an extensive research phase took place.

2. *Design and development stage*

   First steps toward designing an initial architecture with iterative prototyping, as well as the elaboration and delivery of a Project Critical Review document consisting

of any possible changes to this preliminary planning, and ultimately the consolidation of the product by performing a refined training and conducting a thorough performance evaluation of what can be considered the definitive result.

3. *Final stage*

   I assisted in maintaining the integrated model, wrote the report and prepared the defense of this project.

All of these stages and tasks are summarized in this Gantt diagram:

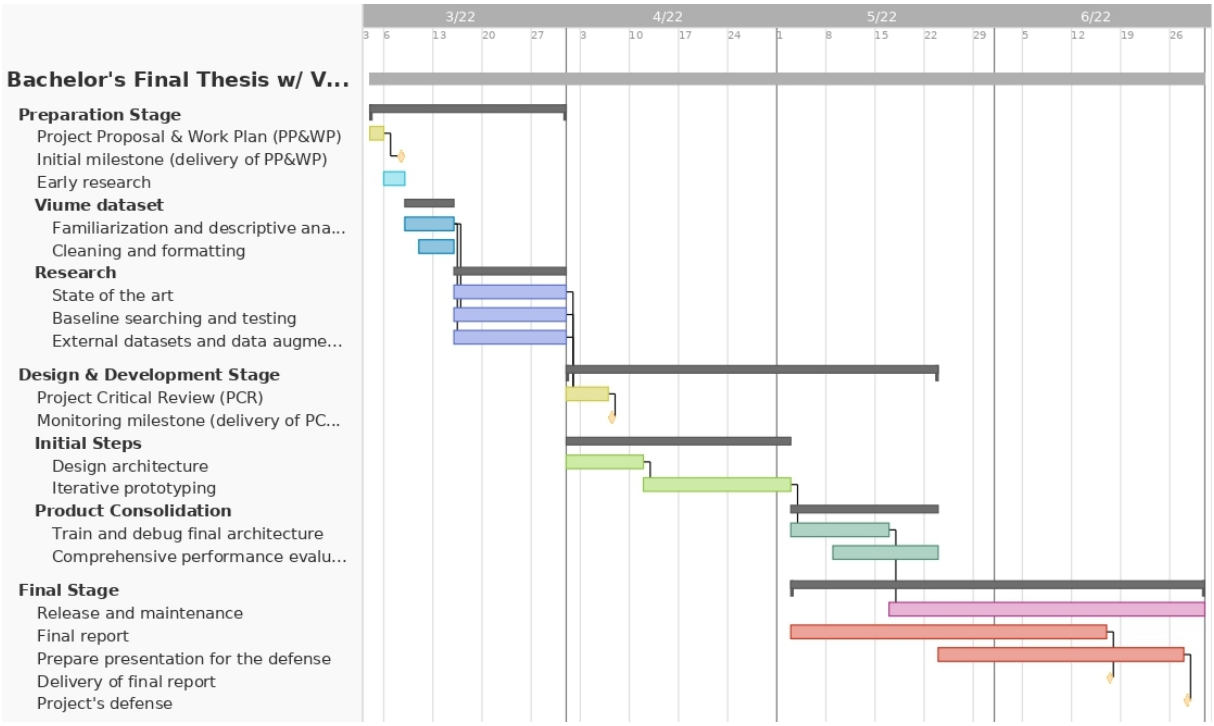


Figure 1.1: Gantt diagram of the project planning.

1.3.1 Amendments to the initial planning

Certain unforeseen incidences have appeared so far during the development of the project, thus entailing changes to the work plan that was initially proposed. In this section, I will explain these incidences and how they affected the planning. Finally, I will present the amended work plan which will serve as the final road map until the completion of the project.

Incidences

The project started with minor incidences concerning the dataset that Viume, the enterprise in which I am carrying out this project as an internship, was supposed to deliver at the start of the project. However, owing to the need to formalize the contract and confidentiality agreement, this delivery was hold back for some weeks. Due to this

delay, it was decided that the research phase would be moved forward until the dataset could be transferred to me, even though ideally they would have been ready from the start to allow me to familiarize with the data and therefore be more aware of what to look for during the research. Moreover, when the dataset finally arrived, it turned out that the data had only gender annotations, and with fairly easy images. As a result, I had to look for alternative sources of data to train a system capable of performing on a in-the-wild context. Thus, I resorted to using datasets like CelebA [35] and MOT17 [38] (from MOTChallenge benchmark [26]). Nevertheless, it is important to note that these datasets have an academic license, meaning that they cannot be used for commercial purposes. Therefore, it is necessary to distinguish between the contents of the thesis and the product that will be finally delivered to the enterprise. As a result, these datasets are used freely inside the context of the thesis but, due to legal reasons, the product that will be delivered to the enterprise will only consist of the final architecture (not trained on any of the aforementioned datasets with academic license) and the research done in the thesis.

During the Design and Development stage, despite having a fairly quick design of the architecture, it took considerably longer than expected to start prototyping due to the lack of adequate data and the need to construct data processing pipelines at this stage. These issues could only had been avoided if there were datasets prepared for our tasks. Unfortunately, this was not the case. Subsequently, another unexpected by-product of this project is a pipeline of semi-automated labeling for face and body bounding boxes, and gender and age labels. Even then, I am currently waiting for the curation process to be completed in order to start properly training and testing my models, since up until now only a small sequence was available, and the models easily overfit on it. The prototyping has been extended up to the end of May, and so the activities in the Product Consolidation, as well as the Release and Maintenance have been shortened to fit the deadline. In contrast, the writing of the Final Report and the preparation for the defense remain untouched. The writing of the Project Critical Review was also delayed in order to take a more informed decision about the necessary changes to the project plan.

Modified Work Plan

Following the above-mentioned incidences, the following modifications to the work plan were adopted:

- Viume dataset and Research phases have switched; Research was done before Viume dataset.
- The delivery of the PCR was postponed until the situation became clearer.
- Iterative prototyping was massively extended to account for the various difficulties encountered during the implementation of the prototypes.
- Given the change on the Iterative Prototyping, Product Consolidation activities were delayed and shortened.

Consequently, this is the final Gantt diagram of the project:

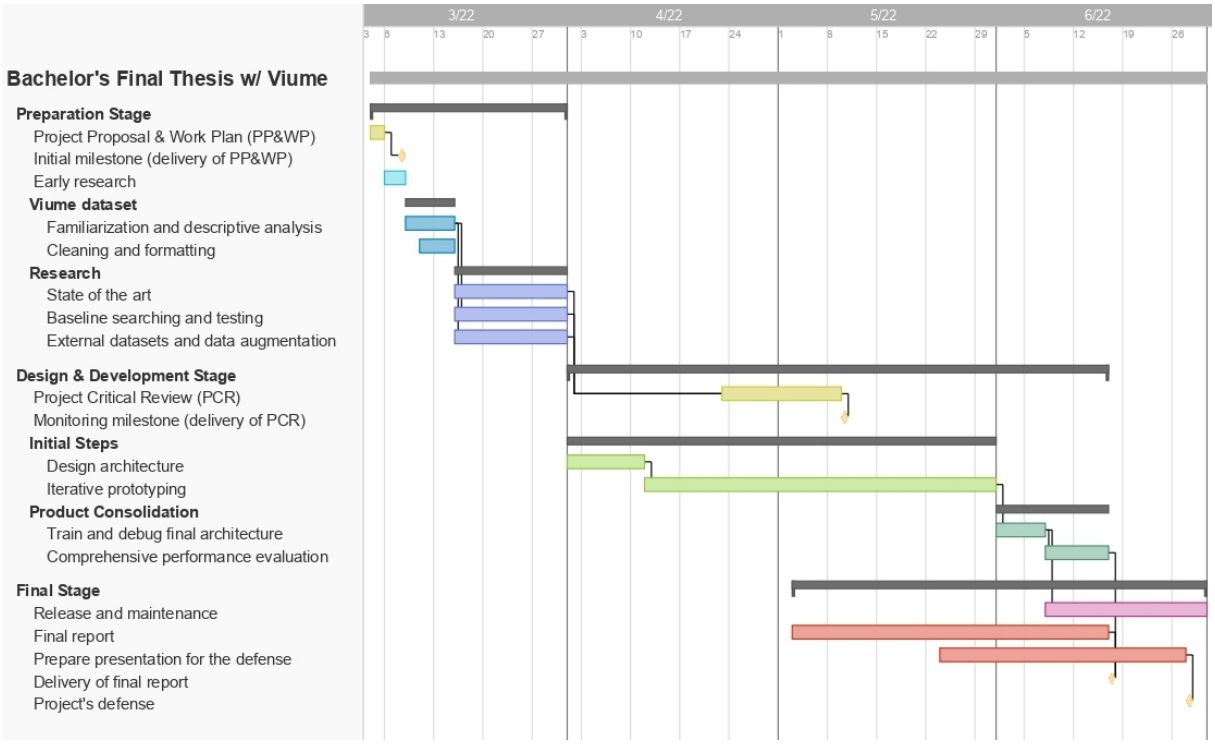


Figure 1.2: Gantt diagram of the modified project planning.

Chapter 2

Related Work

In this section I am going to set the foundations upon which I have developed this project. The first subsection will cover certain essential concepts concerning subjects related to Deep Learning in Computer Vision, the basics of detection and classification in an image data context, and data augmentation for Deep Learning in general. The following subsection will then look into the latest advances for fields closely related to my project, such as full-body detection (normally in pedestrian contexts) and face detection, as well as the state of the art in image feature extractors, architectures and data augmentation specifically designed for the tasks of age and gender recognition.

2.1 Fundamentals

To to understand this project, it is paramount to set the basis of the current state of the art in Computer Vision. Therefore, it is imperative to start by discussing the essential role of Deep Learning, specially in terms of the addition of deeply-learned features from image data, outperforming handcrafted and shallow-learned features in most circumstances. It is mandatory to focus on feature extraction as it has been an essential aspect of the image classification and object detection pipelines, even before features were extracted using neural networks. From this point, we can focus on the detection of objects and image classification using neural networks, introducing work that will be referenced later on when we focus on performing these tasks on our particular case. Finally, given that these types of models are notorious for requiring extensive and diverse datasets, I find it fitting to explain and motivate the need for data augmentation in general terms –in the next subsection we will study how to optimize said augmentation strategies for our own specific kind of data.

2.1.1 Deep Learning in Computer Vision

Deep Learning has revolutionized the world of Computer Vision since its widespread adoption in 2012, so much so that the field has progressed as much in 10 years than it did with the preceding 50 years.

To quickly summarize the approaches of the pre-Deep Learning era of Computer Vision, the first attempts at creating a system able to mimic the same capabilities as human visual perception were based on taking a direct approach to perceiving and analyzing patterns. Perhaps the most iconic of these are Template Matching [39], Histogram Matching [3], and Histogram of Oriented Gradients [7], but these global appearance-based methods had difficulties with changes to pose, illumination, occlusions, etc. Local features were successful in solving this by breaking down the task into recognizing keypoints –parts of the object that characterize it–, as proposed in Feature Matching using Scale-invariant Feature Transforms [30]. Also a very popular method was the Viola-Jones algorithm [59], which used several Haar-like features, mainly to detect faces.

Currently, though, most if not all approaches are based on Deep Neural Networks, and in particular Convolutional Neural Networks (CNN). These allow us to use deeply learned features which are the representation of visual patterns that the network is automatically trained to recognize in order to minimize a loss function. This paradigm was proposed in [24] with the introduction of AlexNet, a CNN that handedly outmatches other contemporary approaches on the ImageNet dataset. Needless to say, all succeeding research was based on this result, first evolving to make the networks deeper, like VGG [50], then de-sequentalize them by branching off from the main flow, like GoogleNet [55], and finally they tried models that reintroduce previous activations later on in the network, the so-called residual networks such as ResNet [15], thus helping the gradient propagate more easily through the network.

Architectures aside, an equally important concept was the use of regulating factors to avoid overfitting. One such method is *dropout* which, as proposed in [52], consists on randomly "dropping out" (nullifying) some of the connections between layers, making it harder for the neural network to rely on sample-dependent features to recognize individual data samples, and therefore aiding it to learn features that can be generalized to other samples. Another important invention is the addition of weight regularization to the loss functions [5], and batch normalization [20] which, apart from having a regulatory effect on the network, also reduces the internal covariate shift and instability in distributions of layer activations and decreases the effect of weight initialization, leading to improvements in training time and accuracy of the neural network.

2.1.2 Feature Extraction

As mentioned earlier, one of the game-changing effects that Deep Learning has had on Computer Vision was the ability to extract rich features from image data. This feature extraction is present in most if not all Computer Vision approaches [60, 32, 12, 11, 45, 44, 31] as it allows us to encode the information present in an image into a vector, usually called *feature vector* or *embedding*. It is therefore used in both of the tasks I will talk about later on –Object Detection and Image Classification– as it allows to translate this visual information into a representation in a feature space. Moreover, as the feature extraction and subsequent tasks are trained together, the network learns to extract features that are adequate for the task at hand. This feature extraction are implemented with CNNs typically referred to as *backbones*. The most classic and popular backbones are AlexNet [23], VGG [50], and ResNet [15].

A revolutionary idea that greatly improved performance in feature extraction was the Feature Pyramid Network [27]. This design pattern leverages a combination of both location rich features from early on in the network and semantically rich features of activations after the dimensional bottleneck, thus obtaining features at different resolutions as well as a more complete description of the image at all these levels.

2.1.3 Object Detection

The task of Object Detection concerns the bounding box localization and classification of objects in images, and is perhaps one of the most research tasks due to the widely differing range of applications. The current cutting-edge detectors can be broken down into two paradigms: *two-stage* and *one-stage*. Of course, the relation of performance between the two is that two-stage approaches are more accurate, but also more computationally demanding than their one-stage counterparts.

One-stage approaches, like YOLO [44], SSD [31], combine both phases of two-stage detectors in a single one by performing a dense sampling of possible object locations, which has the potential to be faster and simpler. Although currently they are still trailing behind two-stage methods, recent research has started to close the gap between the two. Given that these architectures rely on a single look at the image, optimizing the extraction of features as much as possible is paramount to allow subsequent tasks to perform satisfactorily. As explained in subsection 2.1.2, the Feature Pyramid Networks design pattern allows to combine good features both semantically and location-wise, at multiple levels of resolution. The having these properties in object detection implies that we are able to sample proposals of several different sizes and with features that allow to directly classify the object without the need for a second stage. Accordingly, many of the current one-stage approaches use some kind of feature pyramid in their architecture [58, 65, 10]. In addition, since one-stage methods use a densely sample possible object locations, they often have trouble with sparse images where the extreme foreground-background class imbalance encountered during training of dense detectors results in false negatives due to the model's risk adverseness to predicting a minority class. The introduction of Focal Loss [28] helped solve this issue by customizing the cross entropy loss such that the loss from well-classified positive examples is diminished by a focus factor, hence encouraging the model to accept proposals.

Two-stage approaches, typically represented by the R-CNN family of models (from R-CNN [12], to Fast R-CNN [11], and later Faster R-CNN [45]), normally apply a classifier to a sparse set of candidate object locations created using a Region Proposal Network to predict the class of the object and also refine the bounding boxes of the proposed locations. As one-stage approaches will more than certainly always hold a competitive advantage when it comes to speed compared to two-stage, improvements with two-stage have also looked to improve the quality of the predictions. Just as before, FPNs have been a good addition to the main feature extraction stage, although not as impactful as it was with one-stage. However, the focus, rather than on the model itself, has shifted towards other relevant issues –specially in regards to the training process– such as correcting imbalances (e.g. IoU-balanced sampling, balanced feature pyramid, and balanced L1

loss) throughout the architecture [40], changing the region proposal methods [53], and optimizing the training process [62].

2.1.4 Image Classification

The task of Image Classification is the cornerstone of Computer Vision. The first applications of CNNs in Computer Vision was for this task. All implementations essentially boil down to having two stages: to extract the features as commented in subsection 2.1.2, and another one to classify based on the feature map generated by the extractor. This second step was initially done using Support Vector Machines (SVMs) on top, with each SVM trained on the feature map to classify with respect to a single class (hinge loss) [57]. However, this was quickly phased-out in favour of adding a small Multi-Layer Perceptron (MLP) instead of the SVM, popularized by the previously mentioned AlexNet [24]. MLPs have the advantage of allowing to easily scaling up respect the number of classes, and also of jointly predicting all the classes.

2.1.5 Data Augmentation

Up until now, we have analyzed how neural networks can be applied to solve many of the problems. However, an essential requirement for these techniques to work is the availability of large volumes of data, and with enough diversity to represent as much of the target distribution of data as possible. Moreover, considering that our research is aimed at in-the-wild contexts, it is especially crucial for us to ensure our models are robust enough to maintain an acceptable performance even on the least imaginable of scenarios. To address this need, the standard procedure is to implement Data Augmentation in the training process. Data Augmentation refers to the practise of applying transformations to existing data in order to derive new samples. In doing this, we can generate enough new data to avoid overfitting, and even guide the training process into learning the features we are interested in recognizing, although this second option will be explored more in-depth in subsequent sections 2.2.3 and 3.3.

The typical approaches to Data Augmentation are based on performing random transformations such as rotation, changes to hue and illumination, perspective, flipping the image, etc. [49], although since the introduction of Generative Adversarial Networks (GANs) [14], new approaches featuring adversarial transformations can also be found [42, 37].

2.2 State of the Art

Having defined the basics with the summarized background and current research focus with respect to the primary tasks of our project, we move on to see their latest advances. In particular, we will focus on those that work on the same or similar context to ours in terms of the human face and body detection, as well as the recognition of gender and/or age based on visual data, and finally data augmentation techniques designed specifically to aid the training of human detection and classification models.

2.2.1 Latest on Feature Extraction

Feature extraction, a key component of the Computer Vision pipeline, has seen many new approaches in the design of backbones. A new interest in this sub-field was lightening the models while preserving the current performance. In this regard, models like MobileNetV2 [46] and its successor, MobileNetV3 [17], have successfully leveraged what they call the "inverted residual blocks", which are much lighter than the regular residual blocks (MobileNetV2) and then added the "Squeeze-and-Excitation" blocks [18] to explicitly model interdependencies between channels thus helping generalization (MobileNetV3). As a consequence, they maintained a competitive accuracy while drastically reducing inference time. Another family of lightweight models is EfficientNet [56], who are able to slightly diminish the size of the model, while actually even improving the performance by *"carefully balancing network depth, width, and resolution"*, represented by their "compound coefficient". Incidentally, they were able to surpass the famous ResNet family both in accuracy and in having a lower number of parameters.

Approaches using transformers have also seen some success. Although initially being conceived to work in tasks related to Natural Language Processing (NLP), transformers have recently made their way into vision, and these first results suggest that this is a promising approach. The VisionTransformer [9] interprets an image as a sequence of patches and uses Transformer encoder as used in NLP to process them semantically. Furthermore, the fact that it uses such attention mechanisms mean that it is considerably more interpretable than any other DL-based vision model.

Finally, the latest breakthrough in backbone design came out earlier this year. ConvNeXt [34] is a CNN that can be defined as a mix of a ResNeXt [61] and a Swin Transformer (Swin-T) [33], incorporating the grouped and depth-wise convolutions from the former and, from the latter, "patchifying" the convolutions, using inverted bottlenecks and a single activation function for each ConvNeXt block, substituting ReLUs and Batch Normalization layers for Gaussian Error Linear Units (GELUs) [16] and Layer Normalization, as well as a stage compute ratio similar to that used by Swin-T. As a result, they have managed to create a ConvNet that surpasses the latest transformer approaches while staying relatively low on model size, as seen in Figure 2.1. Be that as it may, it is worth pointing out that the researchers of this paper have eluded adding the results of EfficientNet [56], which has quite comparable results in the ImageNet-22K Pre-trained.

2.2.2 Age and Gender Recognition

As discussed in the introduction, nearly all of the state-of-the-art approaches for gender and age recognition have focused on acquiring facial features. Due to intra-class variation of facial images (occlusion, lighting, scale, pose), current models are still under the accuracy level necessary in real-world applications. In order to overcome this, [1] proposes to tackle both tasks at the same time by multitask architecture to jointly predict age and gender, and also to feed the gender prediction to the age classification head. This has shown to increase the accuracy with respect to models trained on separate tasks.

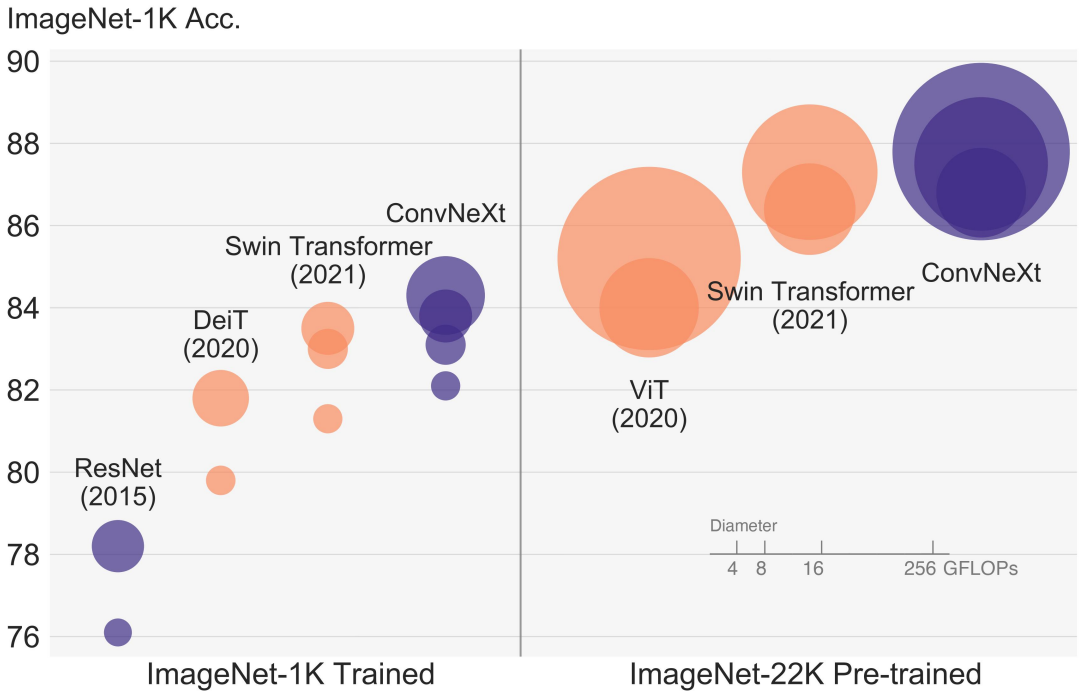


Figure 2.1: Results of ConvNeXt compared to other backbones, courtesy of Liu et al. [34].

Another paper, [21], has looked into more detail how to properly structure MLPs in order to maximize the model's ability to generalize, empirically showing that the appropriate combination of dropout, batch normalization, and skip connections between the fully connected layers can optimize the performance of the model.

Finally, [51] explores the ability of pre-trained networks to adapt to age and gender recognition. By applying their transfer learning techniques, they were able to surpass other state-of-the-art models with a VGG network. Additionally, this paper also infuses a relationship between age and gender predictions by having separate models for predicting age for men and women.

2.2.3 Data Augmentation for Human Detection and Classification

Data augmentation is a delicate when dealing matter with small and sensitive features. A badly designed transformation may produce synthetic data that does not contain the features we might be interested in detecting, or even introduce artifacts, which can lead to counterproductive outcomes, as we would be further shifting the distribution of our training data from the natural data.

For the task of human detection, or similarly pedestrian detection, many attempts at generating realistic data have been tried. As [4] points out, it is harder to synthesize realistic samples from zero than it is from pre-existing real humans. Thus, they propose a framework called Shape Transformation-based Dataset Augmentation which, based on a seed corpus of real instances, it creates augmented data by first running a shape-guided

deformation (basically wrapping the initial pose into another one) on the instance, and then changing the surroundings with the Environment Adaptation module (lighting and hue consistent with context) to finally create a novel data-point. However, other proposals are more in line with directly making the task of the recognition model more difficult during training by introducing "realistic" occlusions and blurring. In the paper [43] in particular, they explore the option of using keypoint detection to specifically target them with occlusions such as blurring and cutouts to minimally distort the images, while attacking parts of the image with important information. However, they conclude that the introduction of rectangle shapes, especially those that seem very much out of place (e.g. black cutouts), can introduce unusual artefacts because natural occlusions can have arbitrary shapes. This is again confirmed by another paper [6], which strongly advocates for mild patch transformations, such as data augmentations in the form of stylized (switching illumination conditions; day to night) and Gaussian augmentations significantly improve the robustness of the model. Other approaches such as KeepAugment [13] simply find the most salient regions and avoid applying transformations to it, although the effectiveness of this approach is debatable considering that we are trying to have a robust interpretability of precisely these regions. In summary, there is quite a lot of research arguing against techniques such as the one proposed in [66], where random parts of the image are substituted by noise, instead supporting the idea of transforming the salient regions of the images without introducing unnatural artifacts that can confuse models trained with them.

As an interesting note, there have been proposals of using adversarial training to have an occluding generator and a classifying discriminator train against one another. Such is the case in [41], where a generator outputs learned optimal transformations, including rotation degrees, scaling and region occlusion, in order to decrease the performance of the adversarial classifier.

Chapter 3

Methodology

The aim of this project is to have a proof of concept when it comes to improving age and gender recognition capabilities in-the-wild, which we mainly attempt to tackle through the addition of features extracted from the body. Hence, the four pillars that constitute the essence of this study are the employed data, the architectures that have been experimented, the training process for these architectures and, finally, the criteria used to evaluate their performance. All of these aspects will be explained in detail in this section.

3.1 Data

Data is at the core of any system based on machine learning. This is especially true when we talk about deep learning, since considerably more data is needed to train neural networks compared to other classic models. What is more, the data also needs to accommodate the tasks we want to train our model on. Along these lines, we do not only require extensive datasets, but also quality data that is fittingly annotated according to the task of age and gender recognition in-the-wild.

Consequently, the ideal dataset for our project would conform to the following specifications:

- Images must contain unconstrained scenarios with foreground individuals at a close to mid distance (around 1 to 10 meters) from the camera.
- Given the requirements for body detection, the data ought to be annotated with bounding boxes indicating the position of the whole body of every person inside the image.
- Given the requirements for face detection, the data ought to be annotated with bounding boxes indicating the position of every visible face inside the image, and linked to the corresponding body bounding box.
- Given the requirements for gender and age recognition, every person with an annotated body bounding box should be classified by gender (into *Man* or *Woman*), and by age (into *Young*, *Adult*, or *Old*).

These desiderata, however, are impossible to obtain in practice, as existing datasets are usually designed with existing paradigms in mind, and therefore new approaches often have to rely on creating their own tailor-made data to suit their niche. As the standard approach to estimating age and gender is facial data, there are many datasets with facial bounding boxes and labels for age and gender. The most noteworthy datasets for these approaches are CelebA [35] and UTKFace [64]. Apart from that, another type of dataset we are also interested in are those for pedestrian detection, as they are comprised of in-the-wild images with annotations for detecting humans which are even identified with ID connecting them throughout the frames for tracking. A prime example of such datasets is the family from MOTChallenge [26] and, in particular, we will be looking at MOT17 [38]. Finally, Viume has also contributed a sampled version of the MMFashion dataset to validate my implementations on a type of data similar to what they are managing.

3.1.1 Adherence to non-commercial research licence

Before delving into the details of the datasets, the reader might be wondering why some of them are being considered when this project is being developed as an internship in an enterprise, while these datasets have a non-commercial research licence. Hence, I find it relevant to provide clarification regarding their usage and how it will affect all parts involved.

It is important to make a distinction between the thesis and the product received by the enterprise. The former is purely academic, invested solely on confirming or disproving the hypothesis that we introduced up until now. The latter will be comprised of all the software and models (with untrained parameters, or trained with data from the enterprise) implemented throughout the internship. Implicitly, these datasets will not contribute to the enterprise in any shape or form.

3.1.2 Employed Datasets

As explained in Section 3.1, four datasets were initially considered: CelebA, UTKFace, MOT17 and MMFashion. However, in the end, the first two were set aside because of their unsuitability regarding their lack of body annotations and especially the context of the images since, as we see in Figures 3.1 and 3.2, their images are often of high quality and with the faces visible, specially UTKFace. Annotation-wise, CelebA is composed of 40 binary attributes, including Man/Woman and Young/Not-Young as well as facial bounding boxes, while UTKFace's annotations consist of age, gender, and ethnicity, in addition to landmarks, but no facial bounding box. In contrast, MOT17 provides street data with pedestrian annotations (body bounding boxes and track IDs), despite not having age and gender labels, which is something we tackle in Section 3.1.3. The MMFashion dataset (see samples in Figure 3.3), as it does not contain any labels, will only be used to visually confirm that the models are able to generalize to out-of-distribution datasets, without playing a major role in the final results. Therefore, the main data I have worked with is that from MOT17, and the extra customizations I have performed on it.

Figure 3.1: Sample images from UTKFace dataset.

Figure 3.2: Sample images from CelebA dataset.

Figure 3.3: Sample images from MMFashion dataset.

MOTChallenge

The Computer Vision community relies on having centralized benchmarks to set standards for the evaluation of performance in multiple research fields. Similarly, MOTChallenge Benchmark's objective is to *"pave the way for a unified framework towards more meaningful quantification of multi-target tracking"*.

The benchmark consists of several tracking sequences, such as the ones shown in Figure 3.4, that depict pedestrians in public spaces. However, since we are interested in a model that would work mainly with close-range images with flat perspectives, we are not interested in using sequences such as the one in the top-left corner, but rather those that are closer to ground view, especially if they are in-doors such as the sequence in the top-right corner. As a consequence, 9 sequences are kept out of the 14 sequences in the original dataset.

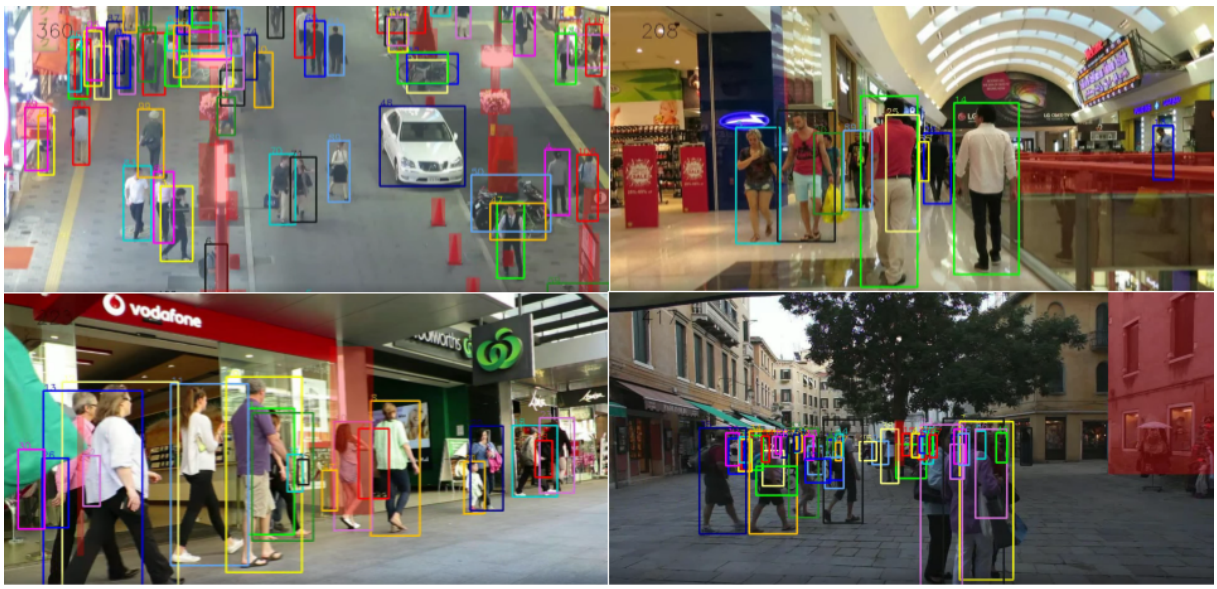


Figure 3.4: Sample images from MOT17 with ground truth detections.

The ground truth file contains the annotations with the following format:

```
<frame>, <id>, <bb_left>, <bb_top>, <bb_width>, <bb_height>, <conf>, <x>, <y>, <z>
```

We are only interested in the frame identifier (`<frame>`) and the bounding box (`<bb_left>`, `<bb_top>`, `<bb_width>`, `<bb_height>`) since we are not doing tracking (no need for `<id>`), nor 3D detections (no need for `<x>`, `<y>`, `<z>` coordinates, which are -1 in 2D challenge ground truths anyway). Indeed, we still need additional annotations for face bounding box, age, and gender. In Section 3.1.3, this issue is addressed with a semi-automatic labelling pipeline that allows us to complete the requirements for the annotations with considerably less effort than would be necessary if we were to annotate the data from scratch.

In essence, and without going into much detail, this semi-automatic labelling pipeline adds a facial bounding box, a boolean value representing 1 for *Man* and 0 for *Woman*, and a value in {0, 1, 2} representing *Young* (approx. 0-20[1] year olds), *Adult* (approx. 20-60[1] year olds), and *Old* (approx. 60+[1] year olds) to a pre-existing detection. In doing so, we obtain a population of whose gender classes are evenly distributed (see Figure 3.6) but, as can be seen in Figure 3.7, the distribution of the classes of age is heavily biased towards the *Adult* label, probably due to it representing a range of ages that, compared to the other two values, correspond to many more people. This is, of course, something I had to work to correct with regards to the training of the model to avoid it developing a bias towards predicting the majority class (see Section 3.3), but also to provide a fair evaluation of the model's performance by looking at the Mean Class Accuracy for ages (as explained in subsection 3.4.3), rather than the overall age accuracy. Additionally, looking at the joint distribution of age and gender (see Figure 3.5) it is worth noting that in the *Old* population there are more than 3 times more women than men. This might incur another bias if the model learns that when predicting *Old* it has a high chance of being correct by predicting *Woman*. This possible bias in the predictions will be studied in more detail during the training process, as explained in Section 3.3.

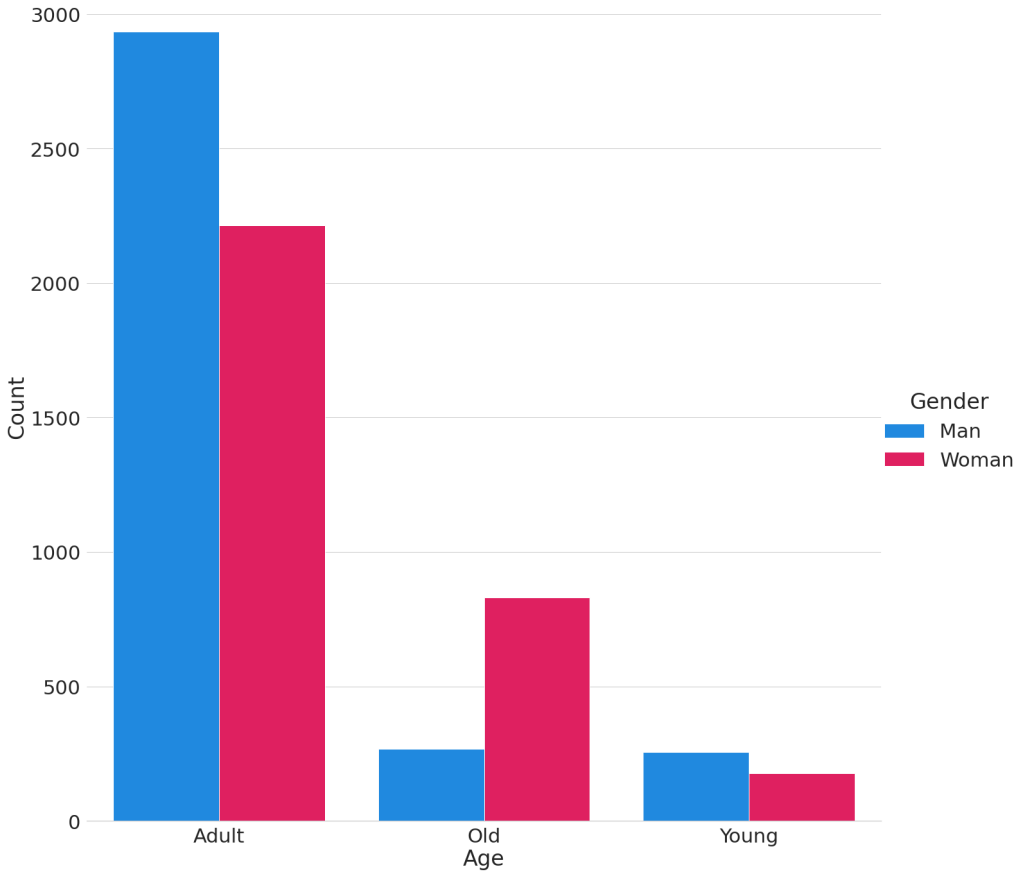


Figure 3.5: Co-occurrence frequency counts between Age and Gender.

[1]The decision to divide the ages into these bins was taken by the enterprise, who took part in the labelling process.

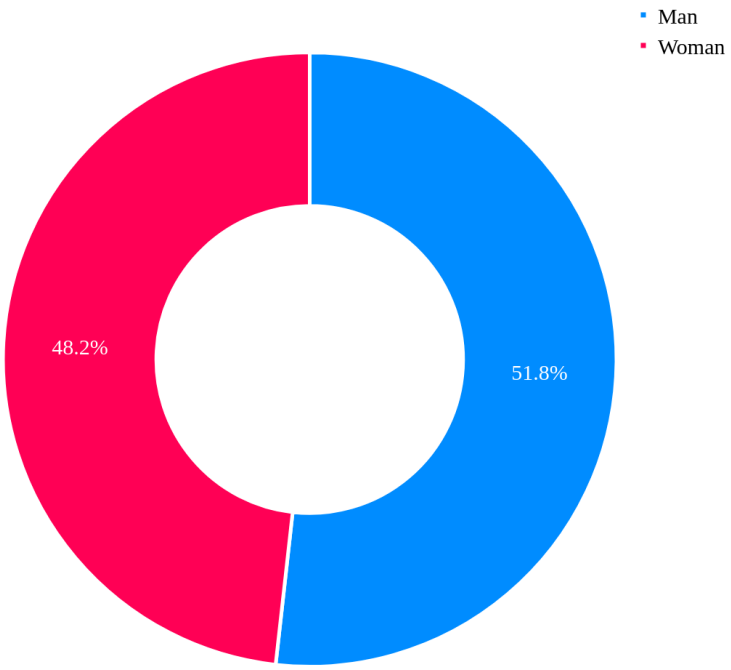


Figure 3.6: Distribution of Gender.

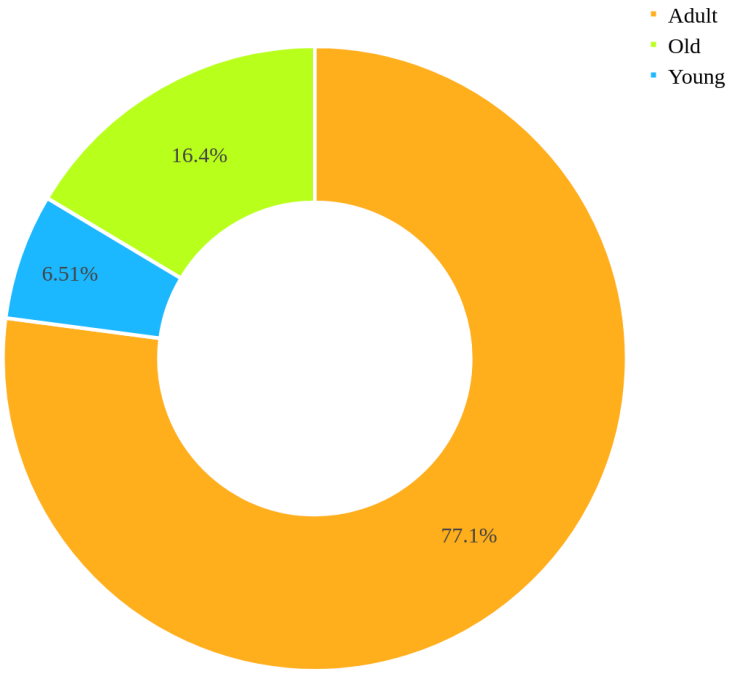


Figure 3.7: Distribution of Age.

3.1.3 Data Transformation Pipeline

The data extracted from the MOTChallenge has to be subjected to several transformations before being ready to be used. There are three main stages in the pipeline: pre-processing, where data is reformatted and properly configured inside the project's working directory, semi-automated labelling, in which the original ground truths for the sequences are extended to accommodate our needs, and finally data loading, where data is augmented (during training) and arranged into a PyTorch-compatible format. The following subsections will dive into more detail for each one of them.

Pre-processing

Given that datasets are downloaded to an unknown location and in different formats, the first requirement for our pipeline is to make the data readily accessible for the rest of the process. As a result, the `pathset_creation.py` script can dynamically create multiple pathsets[2] with minimal interaction from the user. Another great benefit of having this process is the standardization of the data structure, which facilitates the rest of the pipeline. Therefore, a pathset will contain an `imgpath.txt` file, which contains the paths to all the images of the dataset, and an `annotations` directory, which will initially only contain the existing ground truth data.

Semi-automated Labelling

The main objective for this part of the pipeline was to generate the missing labels with the minimal effort possible. Since creating them from scratch would have been an extremely costly affair, our approach consists in taking state-of-the-art models and applying them to generate a first approximation of the labels for us, which we can then curate more easily.

In particular, we require a model to detect the faces, a model to estimate the age and another one to estimate the gender. Perhaps one of the best frameworks out there that combines all of these tasks is DeepFace [48, 47]. Of course, I am unable to provide a numerical evaluation of its performance given that I am using it precisely to generate the ground truth, but in Figure 3.8 there are some images visually showcasing its potential. As can be seen, most of the faces one would expect to be detected are accounted for, but when it comes to the prediction of age and gender, we can see the clear deficiencies that face-based models have in unconstrained contexts, let alone all the people that were not predicted simply because their face was not showing. Even when faces are visible, we can see that those at longer distances in the top two images are sometimes classified incorrectly.

After filtering out the ground truth detections that were not assigned to a human, I use the overlap between the body and face bounding boxes to set the correspondence between the ground truth and the synthetic annotations, and the pairing is done with my own implementation of the Hungarian algorithm [25] based on IoU to decide the pair-up.

[2]Pathset: dataset containing the paths and annotations, but not the images.

Since many of the people do not have a visible face, ground truth bounding boxes that have no pair spawn a datapoint whose face bounding box is null, and placeholder labels are set to age and gender so that later during the curation process they can be changed into the appropriate values.

Finally, the synthetic data is sent to Viume's annotators who, after performing temporal downsampling of the sequences to reduce workload and the redundancy of the data (MOT17 sequences have high fps), return the curated data in a different format to the one which the data was sent. Therefore, the last transformation is needed, which will finally deliver a file containing the curated annotations into the `annotations` folder of each sequence. The end result can be seen in Figure 3.9.

Figure 3.8: Some visual results with the prediction of DeepFace.

Figure 3.9: Final curated annotations for sequence MOT17-12.

Data Loading

The final stage before having the data ready for training is to put it as PyTorch dataloader. To do so, we have a prior step using a customized dataset class which is able to take in samples consisting of the path of the image and the annotations, and return the cropped datapoints with user-customized transformations. In our case, we will be randomly changing the brightness, contrast, saturation, hue, and perspective of the images, and randomly rotate and flip them horizontally. In addition to these, and inspired by the papers [6, 43] who advocate for "artefact-less" transformations in key regions, I have also implemented a customized transformation to blur the faces (see Figure 3.10), thus forcing the model to rely solely on body features. In order to minimize the distribution shift between the training data and the evaluation data that might be introduced by this transformation, the user can specify a probability to apply it, so that some images do contain faces. In Figure 3.11, there are some examples to showcase the effects of these transformations.

Figure 3.10: Original image and different levels of blurring.

Figure 3.11: Joint effect of all transformations.

3.2 Architectures

In this section, I detail the design of the architecture, including the point of view from which I will formulate my experiments. The following subsections will explain the distribution of the flow into two branches –one for predictions using the face and another one that bases its predictions on the whole body–, and how their predictions are combined at the end to harmonize a final prediction.

3.2.1 Ensemble Model

The overall architecture involves performing facial and full-body detection, and then age and gender recognition based on them. It is not trained end-to-end, as it would represent a very difficult challenge to train everything properly, so I instead use models specialized in particular tasks. Moreover, the tasks related to face and those related to body are separated into two branches: a Face Branch (see Section 3.2.2), and a Body Branch (see Section 3.2.3). Whenever possible, I try to use pre-trained models that are able to work with our data, as we do not dispose of enough data to train everything. In fact, I have focused my efforts in training a recognition model that could work with body detections in order to provide a proof of concept for the viability of such approaches as a way to increase the robustness. However, since facial features are often very informative of the age and gender of a person, the ensemble model combines the predictions of both branches to (hopefully) provide more accurate and stable predictions.

3.2.2 Face Branch

This module follows the typical approach of face-based predictions for age and gender. Therefore, it is formed by a face detector and models trained on age and gender prediction based on the features extracted from the face detection. Pre-trained models were used throughout the architecture, but especially in the case of this branch since, given the immense amount of research into this approach, there are many state-of-the-art implementations available that enabled me to construct it solely based on one library. This library is facelib[3], and it features all the necessary components to construct the branch: RetinaFace [8] as face detector, and two ShuffleNets [63] as estimators for age and gender.

I have chosen not to use DeepFace because, unlike in the synthesis of data, this model needs to be able to work simultaneously with the other branch, which is implemented with PyTorch, like facelib, while DeepFace is implemented with TensorFlow. This difference meant that in the initial implementation, where DeepFace was used instead, library incompatibility issues made using it unfeasible. Nevertheless, DeepFace should still be used for the synthesis of data, as it has shown to perform slightly better than facelib, especially for street images, and since in this case it acts as a standalone model, there are no issues with compatibility.

[3] github.com/sajjjadayobi/FaceLib

3.2.3 Body Branch

This branch implements the new approach to age and gender recognition in-the-wild. Given that it is a widely overlooked approach, as far as I am aware, there are no available state-of-the-art implementations or baselines with which to compare or construct our own models with. Even then, since for the first step of the pipeline I need to be able to detect humans, it is still possible to find pre-trained models that can be used off-the-shelf. Consequently, I have picked a Faster R-CNN detector trained on COCO [29] as human detector, given that COCO contains humans as one of the classes to detect and classify. I considered that Faster R-CNN would be the appropriate model since our application is not meant to be executed in real-time, and the increase in accuracy directly affects the performance of subsequent tasks. Therefore, the increase in performance is worth an increase in computational cost. Based on this detection, feature extraction and subsequent classification are performed. Here I have experimented with different convolutional networks to extract features, namely EfficientNet [56], ConvNeXt [34], and ResNet [15], as well as two model architectures –the *Baseline* and the *Baseline–* which are explained in the following subsections.

In regards to the prediction heads themselves, they remain unchanged from one architecture to the other. Their composition, as can be seen in Figure 3.12, is based on what I refer to as "MLP Blocks", consisting of an initial Fully Connected layer ("Linear"), a Batch Normalization step (as recommended by [20]), a ReLU (instead of a GELU, since [34] found no real impact when using them, and therefore it is better to go with the standard approach), and finally Dropout (to fight overfitting, as indicated in [52]). The only difference between the age and gender heads is that, while the one for gender has a Sigmoid activation function to give binary decisions (predicts *Man* or *Woman*), the age head does not so that it can freely regress over the numbers 0, 1, and 2, representing *Young*, *Adult*, and *Old*, respectively. Note that, in these architecture diagrams, blue denotes that a component is (fully or partially) trainable.

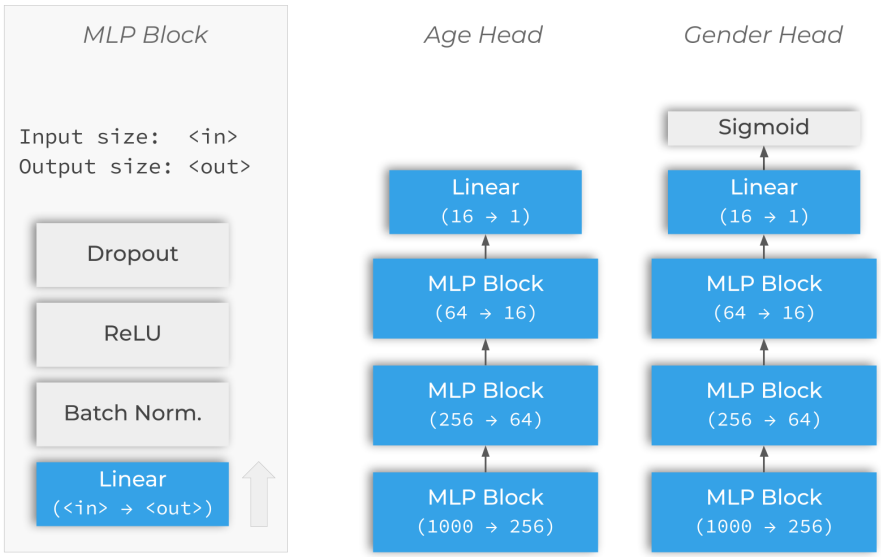


Figure 3.12: Architectures of the age and gender heads, and of the blocks that integrate them.

Baseline

The Baseline is an inefficient implementation, but risk-free performance-wise. It uses two separate feature extractors: one feeding features to the gender head, and another one to the age head. In doing so, we avoid having the potential issues we may find when training joint tasks, as in those cases it is possible that one of the tasks cannot train properly, thus resulting in an underfit. On the other hand, we are duplicating the feature extraction process, hence incurring an overhead in memory usage and computational cost.

In Figure 3.13, the whole ensemble architecture is represented with the Body Predictor being of the Baseline variant. As can be seen, only this part of the architecture will be trained by us, following the reasoning given at the start of this section. Furthermore, since the focus of this project was to provide a proof of concept for the viability of body-based age and gender estimators that could improve on standard approaches for in-the-wild contexts, I find it fitting to focus primarily on this part of the architecture.

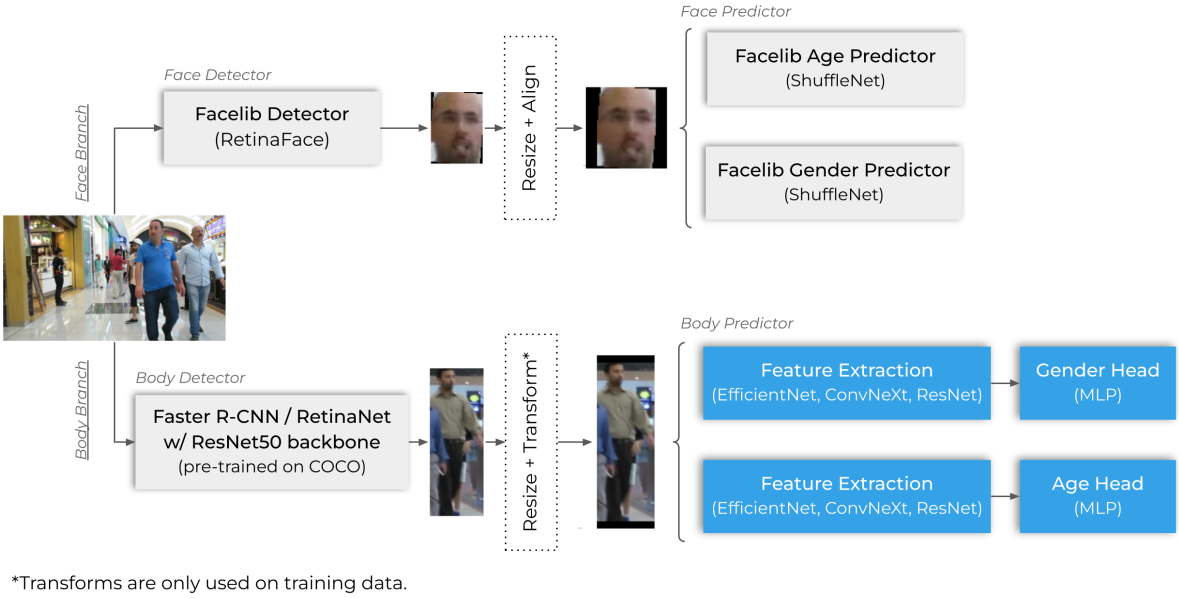


Figure 3.13: Baseline architecture.

Multihead

The Multihead architecture is a multitask approach in which the features of one feature extractor are used by both heads. In doing so, the convolutional network has to learn features useful for both tasks. This is not only positive for the reduction of memory and computation costs it entails, but also because this dual learning can help the model extract more meaningful and generalizable features. Additionally, and according to [1], simultaneous training with this type of multitask networks improves the accuracy for both tasks compared to training them separately.

In Figure 3.13 we can see how the layout changed with respect to the Baseline, as now there is only one feature extractor, which is trained on both tasks at the same time.

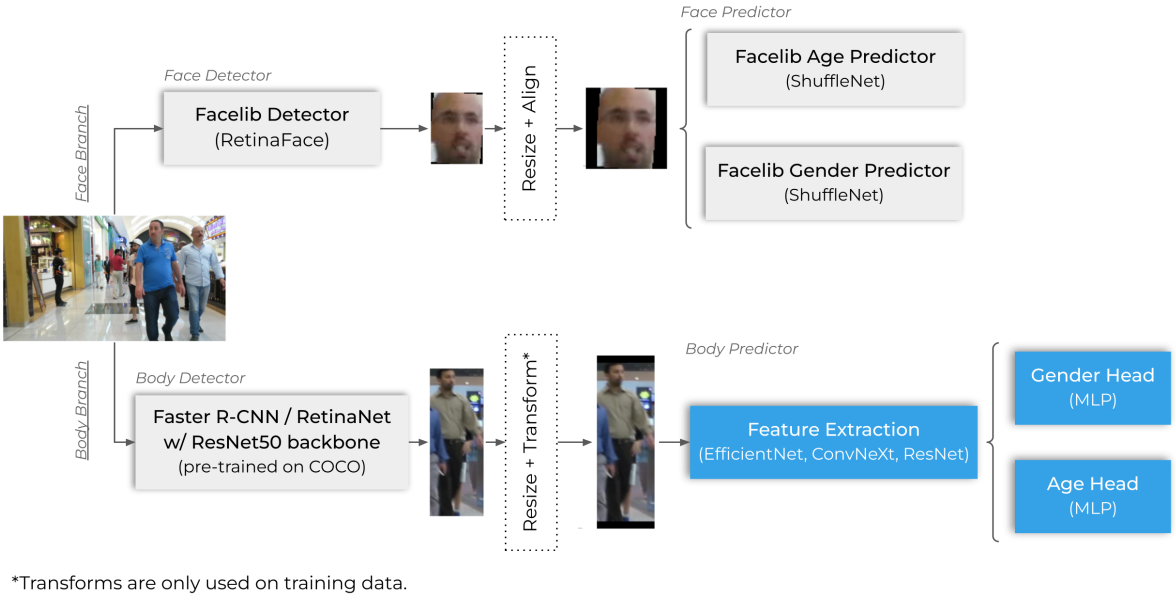


Figure 3.14: Multihead architecture.

3.3 Training Process

The training process, which concerns only the modules drawn in blue, is based on minimizing two loss functions, one for each of the task we are training. To train gender estimation, we are using Binary Cross-Entropy (BCE) loss (see Equation 3.1) given that gender prediction is a binary class prediction problem, and Mean Square Error (MSE) loss (see Equation 3.2) is used for age estimation given that we have modelled the prediction task as a regression problem to preserve the ordering of the age groups.

$$\mathcal{L}_{\text{Gender}}(x, y) = \frac{1}{N}\sum_{i=1}^{N} l_i, \quad l_i = -w_i[y_i \cdot \log x_i + (1 - y_i) \cdot \log(1 - x_i)] \tag{3.1}$$

$$\mathcal{L}_{\text{Age}}(x, y) = \frac{1}{N}\sum_{i=1}^{N} l_i, \quad l_i = (x_i - y_i)^2 \tag{3.2}$$

To minimize these loss functions, I had first tested the optimizers Adam [22] and AdamW [36]. However, experiments found them to be unsuitable to train my model, as training yielded no improvements in performance and gradients were always shown to be close to zero. As it seemed, the optimizer did not update the model's parameters despite seeing high losses. Later on, the classic optimization algorithm Stochastic Gradient Descent was tried, and this time the model managed to train without any issues.

Finally, the sequences has been divided into three splits: Training (∼70% of all detections), Validation (∼15%), and Testing (∼15%). It is difficult to conform the splits to these percentages, especially in the case of cross-validation (see Section 3.4.2), since different sequences have different numbers of detections. However, by picking the largest

sequence for testing, two for validation and the rest for training, these percentages are approximately met.

On top of this foundation, the training process has been carefully designed to manage all the nuances present in our data and architecture in order to ensure the proper optimization of our model. As such, it needs to deal with:

- A relatively small dataset, consisting only of ∼6.6k detections with all sequences –most of which are redundant since we are extracting them from video data.
- Disproportionate distribution of classes for age.
- In the case of Multihead, training both tasks simultaneously without underfitting either of them.

The measures explained in the following subsections were adopted to counteract these issues and mitigate their effect.

3.3.1 Loss Weighting

Existing imbalances in the data can result in the model developing a bias towards predicting the majority class, or whatever label is producing the most loss and is thus driving the training of the model. To fight this overfit, a common practice is to oversample minority classes or undersample majority classes. However, this would require the implementation of a dataloader that, during training, would consider the class proportions of data samples in each batch in order to keep it within approximately equal representation for each class, while having a populational proportionality during validation and testing. On the other hand, one can simply tweak the losses to give more or less importance to certain samples. To do so, we need to only define a weighting parameter with which to multiply the losses. Despite oversampling being a valid option, I decided to only use weighting factors as they are much easier to implement.

As we have seen in Section 3.1.2, our data has a strong bias towards the *Adult* label; 77.1% of the detections are classified as *Adult*. Left unchecked, our model would most likely learn to predict this class most of the time. To restore some importance to the other classes during training, I decided that the best way to weight the classes would be to use the ratio between the proportion of samples from *Adult* and that of the class in question. Therefore, I am weighting *Adult* with a factor of 1 (leaving it as it is), while *Young* has a factor $\frac{.771}{.0651} = 11,84$ and *Old* has a factor $\frac{.771}{.0164} = 47,01$.

Apart from these disproportionalities intrinsic to the data, another issue regarding the difference in magnitude between the losses for gender and those for age could result in one of them driving the optimization process while the other task is left underfitted. Given that the loss for age is an order of magnitude larger than that of gender, I have taken the arbitrary decision to multiply gender loss by a factor of 5 to make them more comparable. I did not multiply it by more since I have observed that for the final epochs the losses start to become more similar, so having this weight be too large may backfire in the later training stages.

3.3.2 Data Augmentation for Training

Transformations are always used to augment the data and thus avoid overfitting on an insufficient amount of data. I have discussed augmentation extensively leading up to this section. As explained in 3.1.3, four basic transformations are used on top of an extra, customized one, face blurring, which helps us guide the training towards only body features. In order to avoid training on data too dissimilar from the validation split, the probability to apply this sequence of transformations was set to 80%. To test their effectiveness, experiments have been done applying all, none, and all except face blurring, in order to show whether faces are relevant or not with our data.

3.3.3 Dropout, Momentum & Weight Decay

The final set of measures put in place to keep overfitting in check use more customary regulatory factors. Dropout, as explained in [52], is a quintessential form of regulating the overfit of a model. As this was shown to not be enough, at the later stages of the project, Weight Decay [5] was also added. Finally, Momentum [54], although not directly related to overfitting, is also used given its properties in speeding up training by avoiding local minima during optimization, especially on loss functions with high curvature.

3.4 Evaluation

To properly assess the performance of our architecture, it is imperative to first clearly define how this will be done, what aspects of the system will be evaluated, and how this evaluation will be carried out, including all involved metrics and experiments. Consequently, we will set the stage for the following experiments by running through these details in the following subsections.

3.4.1 Data Splits

Data needs to be divided into splits in order to evaluate the model in conditions more faithful to those found when it is released. Namely, the majority of the samples will fall under the training split, which is the one that the model will see to perform optimization. However, it cannot be used to provide an estimation of a model's performance, as it is subject to being overfitted, which would result in the model performing very well on this split, but poorly on data that it has not trained with. Additionally, it is also necessary to reserve a testing split, a section of the data that cannot be used to train the model and, in fact, can only be used to evaluate the final model fitted on the training split. Meanwhile, the validation split, despite also being used to provide an "unbiased" estimation, can be used to optimize hyperparameters and decide on an optimal state for a model's parameters.

As introduced in Section 3.3, about 15% of the detections are allocated to testing, the totality of which is conformed by the largest sequence in the dataset in terms of detections, sequence MOT17-09. Then, two sequences have been randomly chosen to conform the

validation split, also of around 15% of all detections. Finally, six sequences are used for training. Sequences are not statically assigned to training or validation since during cross-validation each fold will assign different sequences to the splits. The only constraint is that the percentages of detections for each split remain approximately the same. More information about it is provided in the next subsection.

3.4.2 Cross-Validation

Cross-validation is a useful technique to better estimate the performance of a model, especially when data is scarce. To summarize, it works by generating different partitions for training and validation, called folds, collecting the performance for each one of them, and then averaging it for all the folds. The result is a more unbiased estimation, as the evaluation for individual folds might depend too much on the specific data of that fold, while the average is usually closer to the populational results. Figure 3.15 shows the specific breakdown of the distribution for said folds.

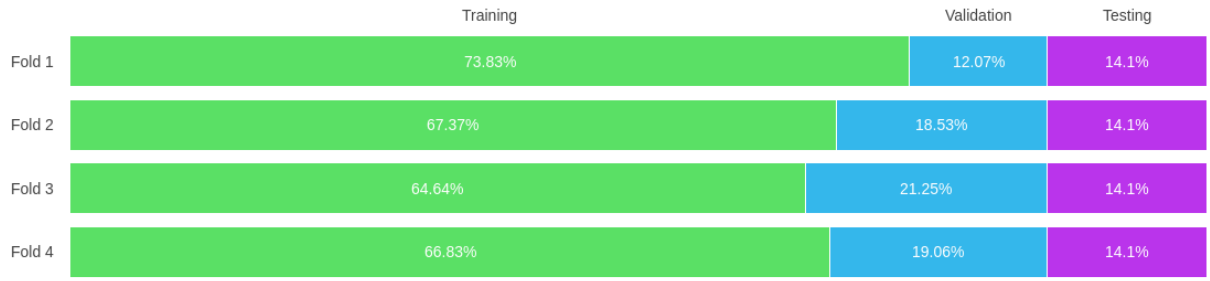


Figure 3.15: Distribution of detections among the splits for each fold.

The testing sequence is always MOT17-09, and the validation sequences are: MOT17-02 and MOT17-05 for Fold 1, MOT17-06 and MOT17-07 for Fold 2, MOT17-08 and MOT17-10 for Fold 3, MOT17-11 and MOT17-12 for Fold 4. The rest is used for training.

3.4.3 Metrics

Using standard metrics to report on the performance of the different tasks is essential for effectively conveying this information. Consequently, this section will formalize the definition of metrics with which to evaluate the performance of detection and prediction tasks.

Detection

Detection metrics are focused around information retrieval, measuring the ability to "retrieve" (i.e. detect) the objects in an image. As such, the employed metrics are *precision* (see Equation 3.3) and *recall* (see Equation 3.4). Additionally, another metric summarizing both precision and recall is *F1 score* (see Equation 3.5) which is a harmonic mean of the two. Finally, Average Precision (AP) (see Equation 3.6) is a metric that was also included given the prevalence in the object detection field. AP@k indicates that we are only considering detections with an overlap of at least k Intersection over Union.

$$\text{Precision (P)} = \frac{TP}{TP + FP} \tag{3.3}$$

$$\text{Recall (R)} = \frac{TP}{TP + FN} \tag{3.4}$$

$$\text{F1 Score} = \frac{2}{\text{P}^{-1} + \text{R}^{-1}} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{3.5}$$

$$\text{AP@}k = \sum_{n \geq k} \left[(\text{R@}[n+1] - \text{R@}n) \cdot \max_{\text{R'} \geq \text{R@}[n+1]} \{\text{P@[R']}\} \right] \tag{3.6}$$

where TP (True Positive) is defined as a predicted detection with enough overlap (Intersection over Union above a certain threshold) with a ground truth detection, FP (False Positive) is defined as a predicted detection that does not have enough overlap with any ground truth detection, and FN (False Negative) is defined as a ground truth detection with no predicted detections overlapping enough with it. The Hungarian algorithm [25] was used to define the bipartite matching between predicted and ground truth bounding boxes.

Gender

The gender recognition task is a binary classification problem. As such, and given that the proportion between both classes is balanced, I find that using the vanilla accuracy, defined as the ratio between correct predictions and total predictions, is an appropriate metric. However, it is also worth pointing out that the performance on this task (and also age estimation) is also affected by the performance of the detection. Therefore, it would also be interesting to know what are the results assuming a perfect detector. This is what I have named the *Oracle Detector* experiment, as it uses the detections from the ground truth instead of the ones provided by the system itself.

Age

Age estimation, despite being modelled as a regression task when it comes to training the model, is in fact a classification between three labels. Given a model's prediction, the nearest integer in {0, 1, 2} is taken as the prediction to be evaluated. As a result, the accuracy we defined for gender could also be used. Having said that, the unbalance in the labels for age may lead to misleading results. Indeed, if the model was to constantly predict the majority class, *Adult*, it would obtain a 77.1% accuracy. Thus, the *Mean Class Accuracy* (MCA) (see Equation 3.7) was taken to complement the vanilla accuracy, as it is a metric similar to accuracy that also keeps in mind the performance over all classes by averaging the accuracies A_i of each class i for all classes in C. Just as with gender, the Oracle Detector experiment is also tested with age.

$$MCA = \frac{1}{|C|} \sum_{i \in C} A_i, \quad A_i = \frac{\#\text{correct}_i}{\#\text{predictions}_i} \tag{3.7}$$

Chapter 4

Experiments

In this chapter, the various aspects of the system's performance will be evaluated through experiments based on the metrics stipulated in Section 3.4.3. Afterwards, in Chapter 5, a subsequent assessment of said results will be discussed as conclusions, and future work proposed.

The aforementioned experiments are comprised of an initial stage of Hyperparameter Optimization for both Baseline and Multihead body branch predictors, followed by a detailed assessment of the Overall Performance, and finally a small experiment where the Effective Range of our approach will be tested.

4.1 Hyperparameter Optimization

The hyperparameters of a model are particularly relevant for its adequate functioning. Hence, their careful tuning is vital to maximize the model's performance. The optimal hyperparameter values for the respective approaches will be established and, as a consequence, so will be the final model for each body predictor approach.

As a means of implementing this process in a timely and appropriate manner, the Sweep functionality of the Weights & Biases Machine Learning framework [2] was used. In addition to configuring and running the tests, the various visualizations that will be shown in this section and the following were also obtained through its logging and visualization functionalities.

The initial approach for both body predictors was to optimize the hyperparameters of Learning Rate, Dropout, Weight Decay and Momentum starting with a random search which would hint towards the ideal range of values for each hyperparameter, and to then perform a grid search to exhaustively probe the reduced search space. The option of a Bayesian search was not adopted given that we are dealing with a multitask problem and we thus have two metrics to optimize, while the Bayesian search requires a single metric to guide its optimization process. It was considered to design a hybrid metric between the accuracies of gender and age –a sort of average between the two–, but it was ultimately abandoned in favour of the approach detailed above.

Something to note beforehand is that, even though MCA would ideally be used as a reference metric for the performance in age, accuracy is used instead. The reason is simply that the initial implementation for age did not consider the possibility of bias. As a result, many of the already executed sweeps would need to be re-run, and the change of implementation itself could lead to complications that are simply not affordable given the time constraints on the project. Therefore, this first part of model optimization does not use MCA, but the later parts where the ensemble model is evaluated do.

4.1.1 Baseline Optimization

The first model in which hyperparameter optimization was performed is the Baseline. As previously explained, the strategy followed to sweep is to perform a random search through an extensive space of possible hyperparameter value combinations with a defined maximum of runs (attempted combinations) to figure out the three best candidate values for each hyperparameter.

Random Search

Random search for the Baseline explored the following search space:

- *Learning Rate*: $\{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$
- *Dropout*: $[0, 0.5]$
- *Momentum*: $[0.7, 0.99]$

The sweeper executed a total of 30 runs, with 10 epochs per run. The results of the sweep indicate that the highlighted ranges are the optimal value combinations of all these hyperparameters (both Figures 4.2, 4.3). It is worth noting that the validation performance in all the runs is considerably lower than that of training. Indeed, despite the measures adopted to avoid overfitting the model has, for the most part, still managed to memorize the training data. To exemplify this incident, we take one of the best performing runs and plot its loss (age loss has been weighted to remove biases) in Figure 4.1.

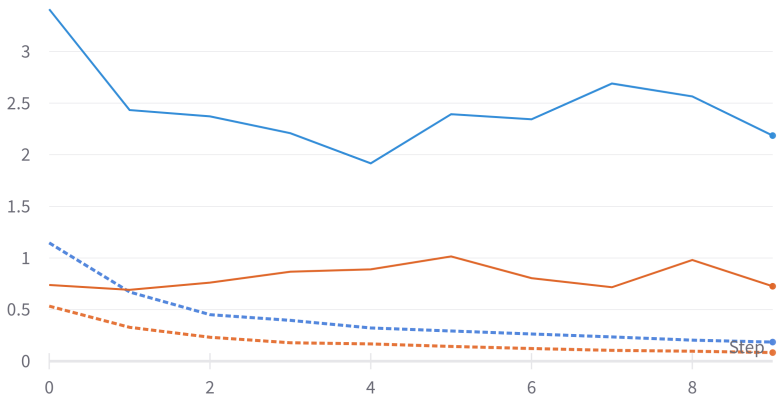


Figure 4.1: Losses during the training process. In blue, age; gender in orange. Validation in continuous lines; dashed for training.

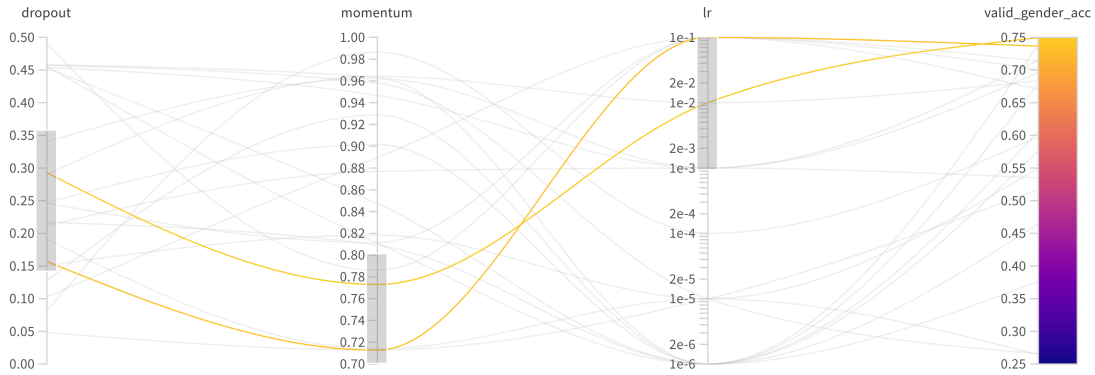


Figure 4.2: Baseline random sweeping with validation gender accuracy.

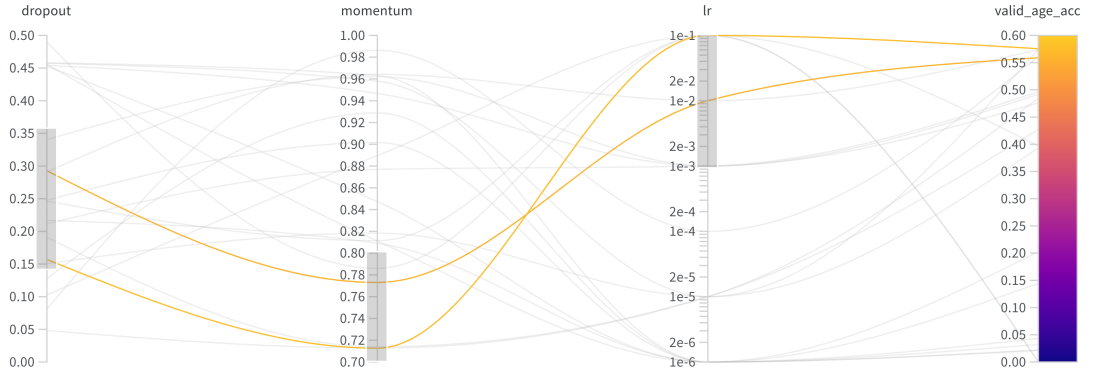


Figure 4.3: Baseline random sweeping with validation age accuracy.

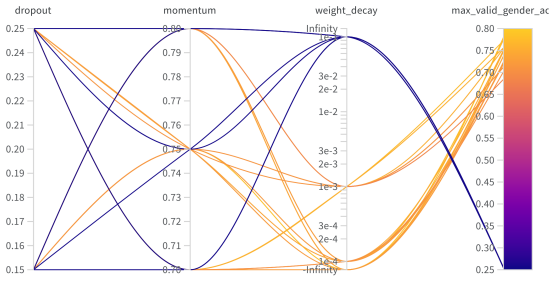
Grid Search

We have reduced the search space based on the highlighted ranges of the random search. Despite that, since the training of these runs has shown that overfitting is present, it was decided that adding weight decay as a last ditched effort to correct the overfit could be worth trying. However, given the time constraints it was not feasible to re-run this sweep, so values that we thought to be appropriate were set instead. Consequently, the reduced search space for the grid search is the following:

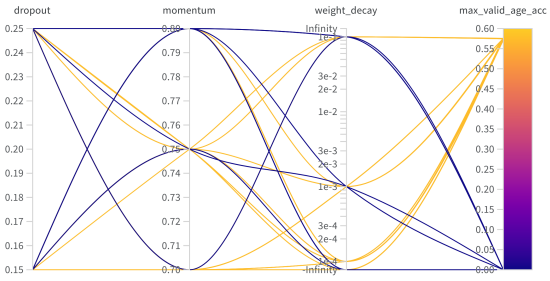
- *Learning Rate*: $\{10^{-1}, 10^{-2}, 10^{-3}\}$
- *Dropout*: $\{0.15, 0.25\}$
- *Momentum*: $\{0.7, 0.75, 0.8\}$
- *Weight Decay*: $\{10^{-1}, 10^{-4}, 0\}$

The grid sweep takes a total of $2 \cdot 3^3 = 54$ runs, almost double than the 30 for random search, which already took around a day to complete. The permissions that I have in

Calcula (UPC's HPC server) do not allow me to run for more than a day and, as a result, the sweep was broken down into three different "sub-sweeps", each with a different value for learning rate. Each one of the Figures 4.4, 4.5, 4.6 corresponds to one of these sub-sweeps.

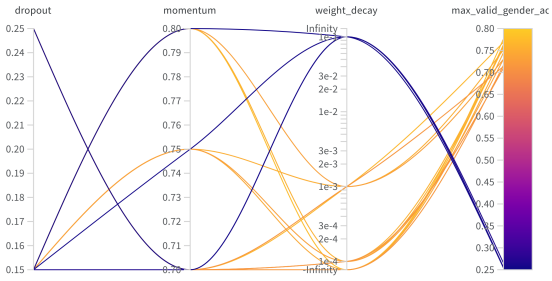


(a) Gender validation accuracy.

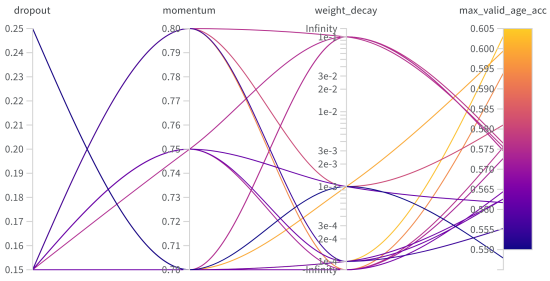


(b) Age validation accuracy.

Figure 4.4: Baseline grid sweeping with LR=10^{-1}.

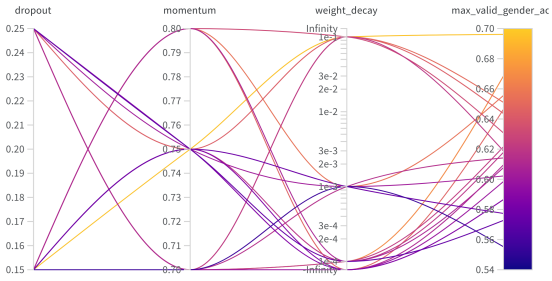


(a) Gender validation accuracy.

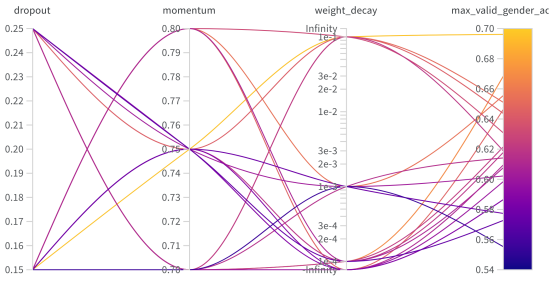


(b) Age validation accuracy.

Figure 4.5: Baseline grid sweeping with LR=10^{-2}.



(a) Gender validation accuracy.



(b) Age validation accuracy.

Figure 4.6: Baseline grid sweeping with LR=10^{-3}.

As seen in all sweeps, the increase in validation accuracy was minimal with respect to the random sweep without weight decay; the Baseline model continues to overfit.

Moreover, all sweeps also indicate that a high weight decay (10^{-1}) is counterproductive independently from the values of other hyperparameters, while lower values have similar performance. Regarding the learning rate, 10^{-1} and 10^{-2} have been confirmed to be the candidate optimal values, as 10^{-3} has produced runs that are generally below those of the other two. Meanwhile, both learning rates of 10^{-1} and 10^{-2} indicate that **0.7 is the optimal value for momentum**. Learning rate 10^{-1} has a run that is optimal in both tasks with a dropout of 0.15, while 10^{-2}'s optimal run has dropout 0.25. Comparing the validation accuracy of these two runs, the former has 77.3% for gender and 57.6% for age, while the latter has 71.0% for gender and 60.3% for age. Therefore, although the latter's performance in both tasks are more balanced, the performance of the former with respect to gender greatly outperforms it. Consequently, **10^{-1} was chosen as optimal learning rate**, along with a **dropout of 0.15** and a **weight decay of 10^{-3}**.

Final Model Cross-Validation

The final model was configured with the hyperparameter optimal values discussed above. As a concluding validation step for these models, cross-validation is used to obtain more accurate performance metrics, given that the validation set used during the sweeps was strongly limited by the insufficient amount of data. Table 4.1 contains the results from training this model with the four folds introduced in Section 3.4.2.

Task	Split	Accuracy (%)				
		Fold 1	Fold 2	Fold 3	Fold 4	Average
Gender	Training	91.98	95.65	88.78	89.06	97.29
	Validation	77.32	72.67	**82.04**	81.75	78.42
Age	Training	69.63	71.41	63.72	60.43	66.30
	Validation	67.64	64.79	77.88	**79.93**	72.56

Table 4.1: Cross-validation results for optimal Baseline.

These results indicate that both tasks have, in theory, been learned with approximately the same proficiency. However, it is best to be cautious and double-check these results with the testing sequence and take MCA to evaluate age, as this will reveal whether the model's predictive behaviour is biased towards the majority class. An unusual incident is that the training accuracy for age is lower than that of validation. The reason for this might be that the validation split for these folds have more bias than the others, thus playing into the hands of a biased predictor. Be that as it may, the state dictionaries for the best performing folds for each task are taken for the corresponding models. These will be used later on during the evaluation of the ensemble model in Section 4.2.

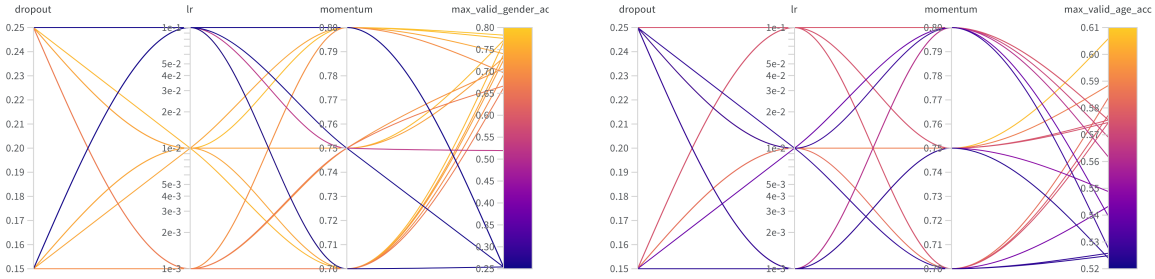
During this cross-validation step, a small experiment was conducted in which transformations were altered or directly removed to check their effect on the model's training. It was shown that **transforming the data increased the validation accuracy, on average for all folds, by 2.3% in gender and 1.76% in age** –not a very significant increase. Furthermore, if only the face blurring was removed, the accuracies were very similar to those with face blurring, thus indicating that it does not have much of an effect.

4.1.2 Multihead Optimization

The optimization of the Multihead had to be done with some modifications to make the process more timely. Starting with the initial random search, the search space for the grid search was determined based on the results found for the Baseline, since it is a similar architecture and time constraints did not allow to do it optimally. Secondly, since weight decay has not had much of an effect in the performance of the Baseline, it was decided to discontinue its experimentation for the Multihead, Consequently, we are only left with a grid sweep consisting on a total of $2 \cdot 3^2 = 18$ runs, which we are able to do with a single execution of the sweeper. The search space for the Multihead sweep is defined having:

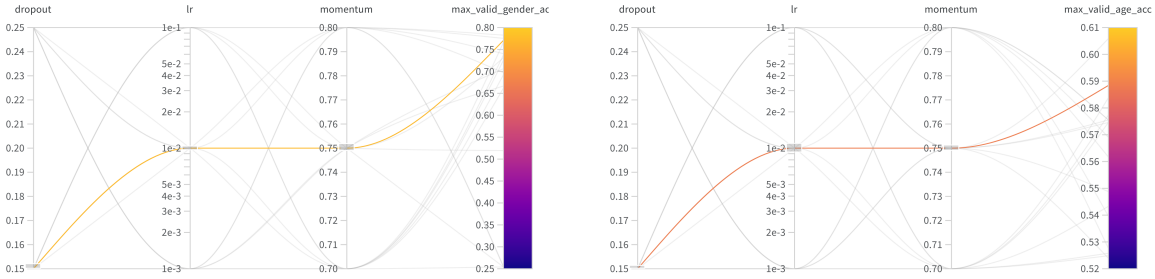
- *Learning Rate*: $\{10^{-1}, 10^{-2}, 10^{-3}\}$
- *Dropout*: $\{0.15, 0.25\}$
- *Momentum*: $\{0.7, 0.75, 0.8\}$

The sweep has generated the results seen in Figure 4.7. In Figure 4.8 we can see the optimal run highlighted, consisting of **0.15 dropout**, **10^{-2} learning rate**, and **0.75 momentum**. This was considered to be the optimal run, as it combines good performance on both tasks.



(a) Gender validation accuracy. **(b)** Age validation accuracy.

Figure 4.7: Multihead grid sweep.



(a) Gender validation accuracy. **(b)** Age validation accuracy.

Figure 4.8: Multihead optimal run.

Final Model Cross-Validation

Similarly to the Baseline, I proceed to train the final model using cross-validation to obtain a more accurate estimation of the model with optimal hyperparameters for Multihead. An important difference between Multihead and Baseline, however, is that since the model trans both tasks simultaneously, one cannot choose an optimal fold for gender and another one for age. Hence, we are taking the one with the highest accuracy on the average of both tasks. Table 4.2 contains the results from training this model with the four folds introduced in Section 3.4.2.

Task	Split	Accuracy (%)				
		Fold 1	Fold 2	Fold 3	Fold 4	Average
Gender	Training	96.42	97.94	92.04	95.87	95.57
	Validation	77.07	74.36	84.86	81.82	79.53
Age	Training	87.88	85.38	89.17	83.76	86.55
	Validation	55.99	65.82	74.84	82.07	69.68
Multitask Validation Average		71.93	70.09	79.85	**81.55**	75.86

Table 4.2: Cross-validation results for optimal Multihead.

Intuitively, it should be harder for Multihead to keep a balanced performance for both tasks compared to the Baseline. Indeed, despite having weighted the loss for age with a factor of 5, Multihead has a 10 percentage point difference between the average validation accuracy of gender and age, compared to the 6% difference seen in the cross-validation table for Baseline. However that may be, we also see that, especially for age, the training's convergence has been better with Multihead's joint training than with the Baseline's separate training, as anticipated by [1].

Repeating Baseline's procedure, it is necessary to determine the optimal state dictionary than needs to be loaded into our ensemble model to evaluate the overall performance in the next section. As commented above, Multihead has optimized both tasks jointly, and the entire state dictionary must be taken as one. Using the multitask validation average accuracy as reference metric to determine the fold to use, the model trained in the fourth fold was chosen.

4.2 Overall Performance

The evaluation of the ensemble model is performed on the testing sequence, MOT17-09, and it includes a Standard Evaluation (see Section 4.2.1), with a subsection for testing detection which consists in measuring the F1 and APs for the detection models, and another subsection for gender and age recognition in which accuracy and MCA are measured for both tasks, and also includingthe Oracle Detector experiment. To conclude, another section with the Effective Range experiment (see Section 4.2.2) checks how the distance to a person affects the capabilities in age and gender tasks by analyzing their performance as a function of a pseudo-distance, a proxy for the real distance given that we lack the ground truth for it.

Before getting to the evaluation, the relevant aspects of the ensemble model need to be specified. The ensemble model combines the predictions of a facial and a body branch: both detect and predict based on their detections independently from one another, and then the ensemble harmonizes these predictions using a weighted sum by a hyperparameter $\alpha \in [0, 1]$ when both are available. This α can be user-tuned depending on the context where it is used: if in a certain context faces are more visible, it might be a good idea to shift the importance towards the prediction of the face branch, while if the people are at a long range, it might be best to rely almost entirely on the body, as faces would not be very visible. In our case, α has been arbitrarily assigned to 0.5. Another aspect are the confidence thresholds used by the detection models. It has been observed that having slightly lower confidence thresholds results in models correctly predicting objects that are further away. However, since they are not included in the curated ground truth, they are counted as false positives. These thresholds were selected by hand to minimize the false positive rates in sequence MOT17-02 (a training sequence).

4.2.1 Standard Evaluation

Detection

Table 4.3 summarizes the detector's performance. The general results for body are quite positive considering that the model is off-the-shelf, pre-trained in COCO [29]. In comparison, performance for face, where an off-the-shelf pre-trained model that was trained on facial detection for age and gender recognition tasks was also used, is awful. A possible explanation for this difference is that, since the images from COCO are more akin to our dataset and facial detectors for age and gender are trained on "easier" datasets, the body detector was more capable of adapting to our in-the-wild context given a relatively smaller distribution shift compared to the face detector. Hence, this exemplifies the issues commented at the introduction relating to state-of-the-art age and gender recognition models being trained on easy and unrealistic contexts.

	Precision	Recall	F1	AP	AP@50	AP@75
Body	.8670	.9559	.9098	.8931	.9338	.8524
Face	.0798	.0684	.0737	.0024	.0083	.0000

Table 4.3: Test set detection results.

If we now focus on the results for body, we can observe that recall is rather high, while precision is somewhat lacking. This was something expected given the issues I have covered in the paragraph above concerning the confidence threshold for detectors. Therefore, these results for body ought to be taken as a lower bound of the actual potential of this body detection model. AP-wise, the results seem nothing out of the ordinary, and are quite comparable to other state-of-the-art detectors as shown in the MOTChallenge benchmark [26], in particular for MOT17 [38]. Indeed, for the detection category, each of the top 3 detectors (BreseeNet [proprietary], Sparse Graph Tracker for detection [19], SeedDet [proprietary]) have managed to obtain an average performance (across all sequences) of 0.9 AP, which is about the same we are obtaining in this sequence.

Gender and Age Recognition

Table 4.4 summarizes both predictors' performance on gender and age estimation using the standard accuracy and the MCA. Additionally, it also includes two columns with the results of the oracle detector. The oracle detector is a small experiment that consists on replacing the model-provided detections by the ground truths themselves in order to simulate the behaviour of a perfect detector. Doing this, we can see how much do the imperfections in our detections affect our ability to predict.

		Baseline		Multihead	
		Own Det.	Oracle Det.	Own Det.	Oracle Det.
Age	Accuracy (%)	66.06	68.76	60.89	62.85
	MCA (%)	33.33	35.44	40.50	41.82
Gender	Accuracy (%)	56.14	57.53	58.24	59.55
	MCA (%)	63.64	67.26	65.41	68.81

Table 4.4: Test set gender and age recognition results.

In general terms, the predictions in the test set have been considerably worse than in validation. Considering that we have already seen many indices of overfitting, this is something that could be expected, but perhaps not at this rate. Recalling some of the previous result in validation from Sections 4.1 and 4.2, we can see that Baseline's gender accuracy dropped from 82.04% to 66.06%, and age's from 79.93% to 56.14%. We see a similar picture with Multihead, having gender go from 81.82% to 58.24%, and age from 83.76% to 60.89%.

Oracle detectors have increased the accuracies by about two percentage points on average. This indicates that, although these detections are good enough to account for most people inside the image (as was seen in the previous subsection), they are lacking some precision in their detections. Nevertheless, this is nothing incredibly worrying, as it would probably be easily solved with some fine-tuning to our dataset.

Exploring the performance on age in more detail, we can see a significant drop between the accuracy that we have been using until now compared to the MCA, hence the importance of using a metric that takes into account possible class biases. In fact, we see that for the Baseline, the MCA is half that of the standard accuracy, while the Multihead is still biased but is seen to perform better with the MCA, in contrast to what the standard accuracy reports. We explore this prediction a bit deeper by visualizing the confusion matrix in Table 4.5, and confirm that there is a clear bias towards the *Adult* class.

		Ground Truth		
		Young	Adult	Old
Predictions	Young	27	53	10
	Adult	63	397	130
	Old	1	23	12

Table 4.5: Confusion matrix for age.

4.2.2 Effective Range

The Effective Range experiment consists in computing the gender and age estimation performance with respect to the distance to the detections. Since this distance does not constitute part of our data, we have to use a pseudo-distance that will correlate to the real distance. To minimize the error of this approximation, we discretize distance into three ranges: *Close* (approx. < 2 meters), *Mid* (approx. 2 to 5 meters), *Long* (approx. > 5 meters).

This proxy distance is modelled as the relative height of the (ground truth) detection compared to that of the frame, which is constant for all frames in all sequences. Hence, we use the ratio of $\frac{h_d}{h_f}$, where h_d is the height of the detection and h_f is the height of the frame. This way, assuming that people have roughly the same height, we can infer that if detection A has a higher ratio than detection B, A will be closer to the camera than B. However, the assumption of equal height would not hold when considering people from different ages, as a nearby child would be classified into a farther distance simply because the person itself is short. Therefore, it is necessary to focus on a particular demographic and, since adults are the majority class, it would make sense to consider only the predictions given to this age group. In order to smooth out further differences inside this class, the distance will also be discretized into the three aforementioned ranges. To do so, two thresholds for the rations have been defined to match the distances defined for the ranges specifically for the testing sequence.

Having defined this system of ranges, the evaluation of the Multihead is repeated (results with Baseline are comparable to these) but only taking into account the accuracy, as we are not interested in the numbers per se, but rather how they change as the distance grows larger, and how an Oracle detection compares to ours, and thus we can ignore the biases. In Figure 4.9 we can observe the results.

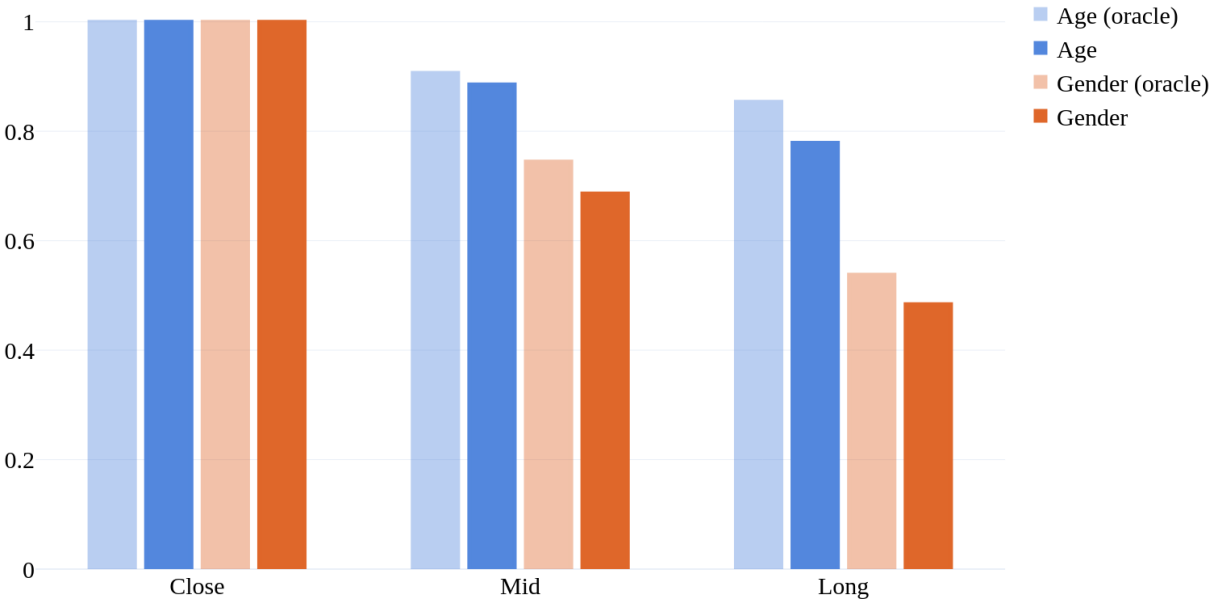


Figure 4.9: Multihead's accuracies for *Adult*-labelled samples as a function of range.

The model has performed surprisingly well in close range, achieving an accuracy of 100%. It would be interesting to know how much of that is due to the face branch, but that is something that will have to be investigated in the future. Also, as commented above, the age performance with samples classified as *Adult* are overvalued when the model has a bias towards this class, but even then the effect of longer distances has decreased it considerably down to the $\sim$77% proportion of *Adults* in the data, although not as much as for gender. In fact, predictions for gender at a long range are basically a 50-50 between the classes. Interestingly, the performance with oracle detectors was always above that of the standard detections, but no important improvements were seen, even for long distances.

Chapter 5

Discussions & Reflections

The conclusion for this project and the future work are presented in this final chapter, where I will discuss the results and findings, and propose continuation routes for the project based on them.

5.1 Conclusions

The evaluation of my approach is conditioned by the fact that there are few papers in this direction, and none that use the same data as me. As such, the conclusions are more inline with what the findings here would mean for future research rather than further detailing the performance of my approach, which was already covered in full in the previous section.

This project has served two main functions: I have empirically shown that face-based detectors and predictors are currently not robust enough for unconstrained scenarios, and that a new paradigm involving the whole body as well as the faces would provide the robustness required. Unfortunately, the overfitting and underfitting on the body-based model prevented me from being able to give a definitive answer on the matter.

5.1.1 Detection

Detection was entirely based on off-the-shelf models, and for body it worked well above my expectations on data that can be considered of as being out of distribution compared to COCO, the pre-training dataset. Certainly, the Oracle Detector and Effective Range experiments have shown that, with some enhancements, performance on the gender and age prediction tasks could be slightly improved. In any case, this issue should not be a primary concern for any future iterations of my approach, as more critical aspects ought to be taken care of before.

On the contrary, face detection has encountered major issues in our dataset. The most likely explanation being the abysmal distribution shift from relatively simple tasks to an unconstrained setting. This comes to show how current state-of-the-art face detection and gender and age recognition frameworks are unprepared for contexts closer to real application settings such as street images.

5.1.2 Gender and Age Recognition

Starting with the comparison between Baseline and Multihead, **the results on the performance itself is rather comparable**, although Multihead seems to have a more stable training, especially with age. What's more, given that it uses the same backbone for both tasks, **Multihead is considerably lighter and quicker than its counterpart**, which makes it more ideal to use in production and, if further optimized, it could even be used in real-time applications. On the other hand, **Multihead also has had significant problems with underfitting age**, as it is hard to keep a balance between the tasks when training them simultaneously.

During the training of the final model for the Baseline in Section 4.1.1, a small experiment consisting in altering or directly removing transformations to check their effect on the model's training showed that these had a minimal effect, in particular face blurring. A possible explanation is that faces are very informative age- and gender-wise, and as a result work very well in contexts where they are well represented. In our dataset, however, this is not the case. In many of the detections, faces have very low resolution or simply cannot be seen, and thus **faces are much less important in an in-the-wild context**. Another argument for the low increase in performance due to data augmentation is that our data can be thought of a natural augmentation of a reduced number of samples: since we are using video data, most of the samples are the same people as other samples except for small changes in resolution (e.g. if they are far and come closer), perspective (e.g. if they turn) and rotation (e.g. if the camera moves), hence why **adding even more transformations to this data has a very limited effect**. The necessary change for our dataset to correct the overfit is to have more separate identities present, rather than creating "new" samples through transformations.

The results of the ensemble model on the testing sequence have shown once again that **the model considerably overfits the training sequences**. Moreover, the MCA has also revealed that **age prediction is noticeably biased towards the *Adult* class** despite my best efforts to avoid it. On another note, the Oracle Detector experiment has shown that the **error due to inaccuracies on the detections are negligible** for now.

Finally, the experiment on the Effective Range has shown that the **performance of our models is significantly reduced by the distance between the person and the camera** –no doubt due to the loss of feature detail due to a lower resolution for objects at longer distances. Furthermore, results for close distances were considerably better, which poses the question whether this increased performance comes mainly from face models having more success with a range more akin to those in their training data, or if it is just an overall increase in both branches. If it were to be the latter, **one could consider getting rid of face models altogether**.

5.2 Future Work

Looking forward, there are many aspects of this work that can be expanded, some of which we have already commented on and motivated. The following list details some of the main courses of action I would consider to have the most potential.

- This project needs more data, and in particular in data that has a wide diversity of individuals and that is not biased towards any demographic.
- A measure to fight underfitting on age would be the previously mentioned oversampling. Thus, it would be recommendable to design a data loader that would balance the training samples across classes of gender and especially age, while keeping the populational biases present in validation.
- Another measure that could help age prediction would be to simplify the regression problem by modelling it as a classification problem instead.
- Certain aspects that needed to be simplified or skipped due to time constraints should be reworked. These include: (i) using MCA to evaluate age during the sweeps and cross-validation, (ii) perform random sweep for the Multihead to build its own reduced search space for the later grid sweep, and (iii) run more combinations for the initial random sweeps.
- I was only able to experiment with the EfficientNet backbone, but it would be interesting to try other state-of-the-art feature extractors such as visual transformers or the recent ConvNeXt, or even more classic models such as ResNet.
- Since we are using video data but only predicting frame-by-frame, a very promising way to increase the robustness of gender and age predictions would be to use tracking. This could be done with a simple implementation using our current frame-by-frame approach and then do a majority voting for the predictions of each track, or something more elaborate would be to extract the features of each detection, summarize the features of each track with a RNN or a transformer into a single feature vector, and then feed it to the classification heads.
- It would be interesting to dig deeper into the Effective Range experiment and test how the facial models hold up against body or ensemble models as distance increases.
- Detectors could be fine-tuned to our dataset, something which could slightly increase our performance in both age and gender.

Bibliography

[1] Amirali Abdolrashidi et al. "Age and gender prediction from face images using attentional convolutional network". In: *arXiv preprint arXiv:2010.03791* (2020).

[2] Lukas Biewald. *Experiment Tracking with Weights and Biases*. Software available from wandb.com. 2020. URL: https://www.wandb.com/.

[3] Peng Chang and J. Krumm. "Object recognition with color cooccurrence histograms". In: *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*. Vol. 2. 1999, 498–504 Vol. 2. DOI: 10.1109/CVPR.1999.784727.

[4] Zhe Chen et al. "A shape transformation-based dataset augmentation framework for pedestrian detection". In: *International Journal of Computer Vision* 129.4 (2021), pp. 1121–1138.

[5] Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. "L2 regularization for learning kernels". In: *arXiv preprint arXiv:1205.2653* (2012).

[6] Sebastian Cygert and Andrzej Czyżewski. "Toward robust pedestrian detection with data augmentation". In: *IEEE Access* 8 (2020), pp. 136674–136683.

[7] N. Dalal and B. Triggs. "Histograms of oriented gradients for human detection". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. 2005, 886–893 vol. 1. DOI: 10.1109/CVPR.2005.177.

[8] Jiankang Deng et al. "Retinaface: Single-shot multi-level face localisation in the wild". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 5203–5212.

[9] Alexey Dosovitskiy et al. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv:2010.11929* (2020).

[10] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V. Le. "NAS-FPN: Learning Scalable Feature Pyramid Architecture for Object Detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.

[11] Ross Girshick. "Fast R-CNN". In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 1440–1448. DOI: 10.1109/ICCV.2015.169.

[12] Ross Girshick et al. "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Nov. 2013). DOI: 10.1109/CVPR.2014.81.

[13] Chengyue Gong et al. “KeepAugment: A simple information-preserving data augmentation approach”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 1055–1064.

[14] Ian Goodfellow et al. “Generative adversarial nets”. In: *Advances in neural information processing systems* 27 (2014).

[15] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[16] Dan Hendrycks and Kevin Gimpel. “Gaussian error linear units (gelus)”. In: *arXiv preprint arXiv:1606.08415* (2016).

[17] Andrew Howard et al. “Searching for mobilenetv3”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 1314–1324.

[18] Jie Hu, Li Shen, and Gang Sun. “Squeeze-and-excitation networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 7132–7141.

[19] Jeongseok Hyun et al. “Detection Recovery in Online Multi-Object Tracking with Sparse Graph Tracker”. In: *arXiv preprint arXiv:2205.00968* (2022).

[20] Sergey Ioffe and Christian Szegedy. “Batch normalization: Accelerating deep network training by reducing internal covariate shift”. In: *International conference on machine learning*. PMLR. 2015, pp. 448–456.

[21] Taewoon Kim. “Generalizing MLPs With Dropouts, Batch Normalization, and Skip Connections”. In: *arXiv preprint arXiv:2108.08186* (2021).

[22] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[23] Alex Krizhevsky. “One weird trick for parallelizing convolutional neural networks”. In: *arXiv preprint arXiv:1404.5997* (2014).

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Commun. ACM* 60.6 (Apr. 2017), pp. 84–90. ISSN: 0001-0782. DOI: 10.1145/3065386. URL: https://doi.org/10.1145/3065386.

[25] Harold W Kuhn. “The Hungarian method for the assignment problem”. In: *Naval research logistics quarterly* 2.1-2 (1955), pp. 83–97.

[26] Leal-Taixé, Laura et al. “Motchallenge 2015: Towards a benchmark for multi-target tracking”. In: *arXiv preprint arXiv:1504.01942* (2015).

[27] Tsung-Yi Lin et al. “Feature pyramid networks for object detection”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2117–2125.

[28] Tsung-Yi Lin et al. “Focal Loss for Dense Object Detection”. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Oct. 2017.

[29] Tsung-Yi Lin et al. “Microsoft coco: Common objects in context”. In: *European conference on computer vision*. Springer. 2014, pp. 740–755.

[30] Tony Lindeberg. "Scale Invariant Feature Transform". In: vol. 7. May 2012. DOI: 10.4249/scholarpedia.10491.

[31] Wei Liu et al. "SSD: Single Shot MultiBox Detector." In: *ECCV (1)*. Ed. by Bastian Leibe et al. Vol. 9905. Lecture Notes in Computer Science. Springer, 2016, pp. 21–37. ISBN: 978-3-319-46447-3. URL: http://dblp.uni-trier.de/db/conf/eccv/eccv2016-1.html#LiuAESRFB16.

[32] Xin Liu et al. "Agenet: Deeply learned regressor and classifier for robust apparent age estimation". In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2015, pp. 16–24.

[33] Ze Liu et al. "Swin transformer: Hierarchical vision transformer using shifted windows". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 10012–10022.

[34] Zhuang Liu et al. "A ConvNet for the 2020s". In: *arXiv preprint arXiv:2201.03545* (2022).

[35] Ziwei Liu et al. "Large-scale celebfaces attributes (celeba) dataset". In: *Retrieved August* 15.2018 (2018), p. 11.

[36] Ilya Loshchilov and Frank Hutter. "Decoupled weight decay regularization". In: *arXiv preprint arXiv:1711.05101* (2017).

[37] Agnieszka Mikołajczyk and Michał Grochowski. "Data augmentation for improving deep learning in image classification problem". In: *2018 international interdisciplinary PhD workshop (IIPhDW)*. IEEE. 2018, pp. 117–122.

[38] Anton Milan et al. "MOT16: A benchmark for multi-object tracking". In: *arXiv preprint arXiv:1603.00831* (2016).

[39] Marvin Minsky. "Steps toward Artificial Intelligence". In: *Proceedings of the IRE* 49.1 (1961), pp. 8–30. DOI: 10.1109/JRPROC.1961.287775.

[40] Jiangmiao Pang et al. "Libra R-CNN: Towards Balanced Learning for Object Detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.

[41] Xi Peng et al. "Jointly optimize data augmentation and network training: Adversarial data augmentation in human pose estimation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 2226–2234.

[42] Luis Perez and Jason Wang. "The effectiveness of data augmentation in image classification using deep learning". In: *arXiv preprint arXiv:1712.04621* (2017).

[43] Rafal Pytel, Osman Semih Kayhan, and Jan C van Gemert. "Tilting at windmills: Data augmentation for deep pose estimation does not help with occlusions". In: *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE. 2021, pp. 10568–10575.

[44] Joseph Redmon et al. "You Only Look Once: Unified, Real-Time Object Detection". In: June 2016, pp. 779–788. DOI: 10.1109/CVPR.2016.91.

[45] Shaoqing Ren et al. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.6 (2017), pp. 1137–1149. DOI: 10.1109/TPAMI.2016.2577031.

[46] Mark Sandler et al. “Mobilenetv2: Inverted residuals and linear bottlenecks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 4510–4520.

[47] Sefik Ilkin Serengil and Alper Ozpinar. “HyperExtended LightFace: A Facial Attribute Analysis Framework”. In: *2021 International Conference on Engineering and Emerging Technologies (ICEET)*. IEEE. 2021, pp. 1–4. DOI: 10.1109/ICEET53442.2021.9659697. URL: https://doi.org/10.1109/ICEET53442.2021.9659697.

[48] Sefik Ilkin Serengil and Alper Ozpinar. “LightFace: A Hybrid Deep Face Recognition Framework”. In: *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*. IEEE. 2020, pp. 23–27. DOI: 10.1109/ASYU50717.2020.9259802. URL: https://doi.org/10.1109/ASYU50717.2020.9259802.

[49] Connor Shorten and Taghi M Khoshgoftaar. “A survey on image data augmentation for deep learning”. In: *Journal of big data* 6.1 (2019), pp. 1–48.

[50] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

[51] Philip Smith and Cuixian Chen. “Transfer learning with deep CNNs for gender recognition and age estimation”. In: *2018 IEEE International Conference on Big Data (Big Data)*. IEEE. 2018, pp. 2564–2571.

[52] Nitish Srivastava et al. “Dropout: a simple way to prevent neural networks from overfitting”. In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958.

[53] Peize Sun et al. “Sparse R-CNN: End-to-End Object Detection With Learnable Proposals”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2021, pp. 14454–14463.

[54] Ilya Sutskever et al. “On the importance of initialization and momentum in deep learning”. In: *International conference on machine learning*. PMLR. 2013, pp. 1139–1147.

[55] Christian Szegedy et al. “Going deeper with convolutions”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9.

[56] Mingxing Tan and Quoc Le. “Efficientnet: Rethinking model scaling for convolutional neural networks”. In: *International conference on machine learning*. PMLR. 2019, pp. 6105–6114.

[57] Yichuan Tang. “Deep learning using linear support vector machines”. In: *arXiv preprint arXiv:1306.0239* (2013).

[58] Zhi Tian et al. “FCOS: Fully Convolutional One-Stage Object Detection”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2019.

[59] P. Viola and M. Jones. “Robust real-time face detection”. In: *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*. Vol. 2. 2001, pp. 747–747. DOI: 10.1109/ICCV.2001.937709.

[60] Xiaolong Wang, Rui Guo, and Chandra Kambhamettu. “Deeply-learned feature for age estimation”. In: *2015 IEEE Winter Conference on Applications of Computer Vision*. IEEE. 2015, pp. 534–541.

[61] Saining Xie et al. “Aggregated residual transformations for deep neural networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1492–1500.

[62] Hongkai Zhang et al. “Dynamic R-CNN: Towards high quality object detection via dynamic training”. In: *European conference on computer vision*. Springer. 2020, pp. 260–275.

[63] Xiangyu Zhang et al. “Shufflenet: An extremely efficient convolutional neural network for mobile devices”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 6848–6856.

[64] Song Yang Zhang Zhifei and Qi Hairong. “Age Progression/Regression by Conditional Adversarial Autoencoder”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2017.

[65] Qijie Zhao et al. “M2det: A single-shot object detector based on multi-level feature pyramid network”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 33. 01. 2019, pp. 9259–9266.

[66] Zhun Zhong et al. “Random erasing data augmentation”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 07. 2020, pp. 13001–13008.