POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

PAGE: advances in integrating pedestrian detection, age, and gender estimation

Tesi di Laurea Magistrale in
Computer Science and Engineering - Ingegneria Informatica

Davide Foini, 987585

Advisor:
Prof. Matteo Matteucci

Co-advisors:
Simone Mentasti

Academic year:
2022-2023

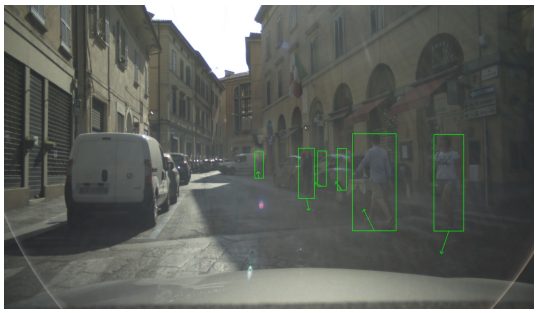
Abstract: Pedestrian analysis is employed in a variety of fields, from security applications to pedestrian traffic analysis or other commercial purposes. To perform this operation, it is first necessary to perform pedestrian detection and then extract useful information. This work presents PAGE (Pedestrian Age and Gender Estimation), which combines detection and age and gender estimation in one lightweight pipeline. It exploits the YOLOX detector and different models to classify body and head images with low resolution. The experiments carried out show how the model can be considered a promising first step towards future developments in pedestrian analysis. The other contribution of this work is a series of tests aimed at improving the accuracy of an age estimation model that needs to classify an image into eight different age ranges. As the baseline model, MobileNet has been chosen due to its good tradeoff between accuracy and model size. Using a regressor or a classifier where each class is an age value has been proven to be significantly better than employing a classifier where each class is an age range, both in terms of performance and flexibility. Other tests showed how the input image resolution and the model size are essential factors when estimating the age of an individual.

Key-words: pedestrian detection, age estimation, gender classification

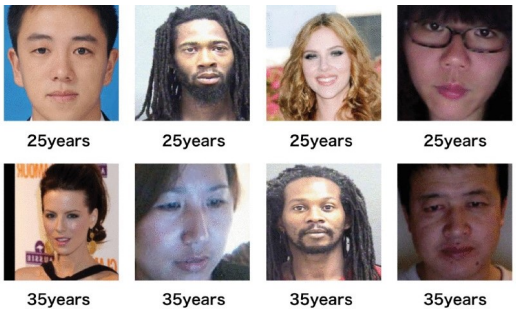
1. Introduction

1.1. Context

The field of Machine Learning (ML) is a subset of Artificial Intelligence (AI) that has undergone significant advancements, particularly in the past decade, thanks to improved hardware computational capabilities. As a result, it has rapidly proliferated across various markets, including healthcare and robotics.
A variety of applications exist for analyzing pedestrians, such as pedestrian detection, which involves locating individuals using bounding boxes in pictures or video sequences. It is a special instance of object detection, where the intent is to locate and recognize different objects. Pedestrian detection can identify the full body, head, or both depending on the application. It is used in fields like autonomous driving, surveillance, and tracking, where the detector plays a central role.
Age and gender estimation can be considered a case of Pedestrian Attribute Recognition (PAR) [60], where the age is commonly estimated from facial images and the gender can be inferred also from the full-body image.

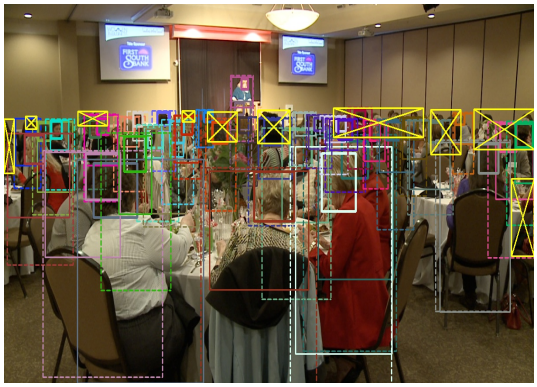


(a) An example of pedestrian detection for autonomous driving taken from [3].



(b) Examples of face images with true age underneath [64]. It is evident from these pictures that individuals of the same age may appear vastly dissimilar.

Figure 1: Examples of (a) pedestrian detection and (b) age regression.



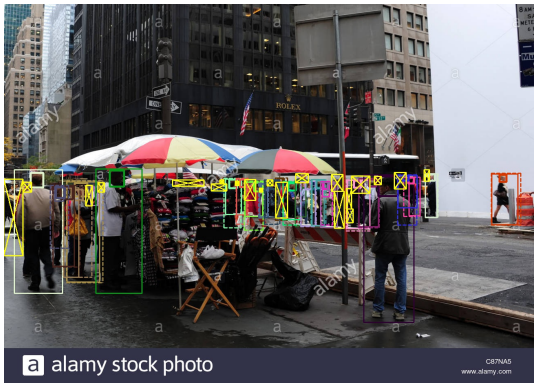


Figure 2: Examples of crowded scenes with annotations taken from the CrowdHuman dataset [46], where the dotted-line boxes are the full boxes, the full-line the visible ones and crossed boxes are occluded.

Most of these applications are designed to be used on mobile devices, mainly due to the physical requirements of the settings and for privacy reasons. Onboard computing is preferred as it does not involve the transmission of sensitive data. However, current research on these topics often overlooks the hardware limitations.
One of the drawbacks of the current age estimation models is that they are usually trained and tested on a single dataset. As a result, their performance tends to suffer when tested on other datasets with different layouts and settings. This highlights their lack of generalization capabilities. Moreover, most of the public datasets available are unbalanced in the age range 25-35, therefore while the performance shown could be satisfying on average, the accuracy on lower and higher age ranges could be significantly worse.
Having considered these aspects, this work provides two contributions. The first is PAGE (Pedestrian Age and Gender Estimation), a framework that combines pedestrian detection, age regression and gender classification, intended to run online and on a mobile device. The second contribution is to propose a model that performs age classification based on age groups, always meant to run on an embedded device.

1.2. Challenges

Performing pedestrian detection in public spaces can be challenging, especially when there are high crowd densities. This can cause significant occlusions and overlaps, which can reduce the performance of the models and slow down the processing rate, especially when dealing with the limited resources of mobile devices.
Besides the lack of both younger and older age groups, when performing age estimation one of the main challenges is intra-class variation. This means that people of the same age may have physical differences, which makes it harder to recognize common characteristics. This is clearly visible in Figure 1b.
Moreover, age estimation can either be approached as a regression problem by directly estimating a subject's age or as a classification problem, where each class is an age value or group. Different studies chose which path to follow.

1.3. Project Structure

This project is structured as follows. In Section 2 the current state-of-the-art of pedestrian detection and age and gender estimation is analysed. In Section 3 the PAGE framework is described, from its architecture to its run-time functioning and the experiments performed. Section 4 reports the models developed to classify facial images based on different age groups and the tests carried out. This work ends with Section 5, where the main results and contributions are summarised and possible future developments are discussed.

2. Related Work

2.1. Pedestrian Detection

The early methods that tackled the problem involved the usage of hand-crafted features, such as the Haar wavelets representation [39] or *rectangle features* [56], that represent the difference of the pixel values of different rectangular regions (ranging from *two-rectangle features* to *four-rectangle features*).
Recent studies have focused on the use of Convolutional Neural Networks, to perform general object detection but also pedestrian detection. One of the most popular models is represented by YOLO (You Only Look Once), first introduced in [41], which is a one-stage detector based on a simple CNN (Convolutional Neural Network) that predicts both bounding boxes and classes, that has achieved great real-time performances. After the first model, many more versions were released by Redmon et al. [42, 43] or by other authors, like YOLOv7 [57] and YOLOv8 (for which a paper has not been released at the moment).

In particular, for this work, the YOLOX [14] version has been chosen as the foundation of our pipeline. It is based on the previous version YOLOv3 [43], but with some modifications. The first major modification is that YOLOX is composed of a backbone and a FPN (Feature Pyramid Network) [29] like the previous models, but the second stage of the network is structured as a *decoupled head* instead of a single head, where one head is responsible for classifying the detections and the other of computing the bounding boxes. The other important change is the switch from *anchor-based* to *anchor-free* detection. Anchors can be defined as predefined bounding boxes that are tiled on the input image to detect objects and their size needs to be determined to achieve a better performance. For this reason, anchor-based detectors are domain-specific and less able to generalize. Removing their usage has also made the architecture more simple and lightweight, improving the real-time and on-device performances. Different models are proposed with various sizes, starting from YOLOX-nano and YOLO-tiny for mobile devices to YOLOX-S, YOLOX-M, YOLOX-L, and YOLOX-X which have increasing sizes. YOLOX has been tested on the COCO dataset and it has obtained an average precision (AP) ranging from 24.3% with the nano version to 51.2% with the X version. The Average Precision (AP) formula involves calculating precision and recall for different IoU thresholds and averaging the results. The piecewise function for the precision-recall curve can be represented as:

$$\mathrm{AP}_{\mathrm{IoU}} = \frac{1}{10} \sum_{\mathrm{IoU}} \mathrm{AP}_{\mathrm{IoU}}, \tag{1}$$

where the summation is over 10 IoU thresholds ranging from 0.50 to 0.95 with a step size of 0.05. The Mean Average Precision (mAP) is then computed by averaging AP over all categories:

$$\mathrm{mAP} = \frac{1}{C} \sum_{\mathrm{category}} \mathrm{AP}_{\mathrm{category}}, \tag{2}$$

where C is the total number of categories. AP and mAP are often used interchangeably. Therefore, if a specific class is not identified when referring to AP, it should be interpreted as mAP.

Another work based on the YOLOv3 architecture is RSA-YOLO [22], which stands for Ration-and-Scale-Aware YOLO. Many studies compress the input images to a fixed size, possibly causing distortions. This is due to the different aspect ratios of the input images and can lead to poor performance. RSA-YOLO aims to mitigate this effect. To do so, two main components are used: ratio-aware YOLO (RA-YOLO) and a multi-resolution fusion module. The ratio-aware mechanism dynamically tunes the hyperparameters of the network based on the input image changing the input layer parameters based on the given image. The hyperparameters of the input layer

w and h can be computed through a set of equations:

$$
\begin{aligned}
w &= \begin{cases} \gamma, & \text{if } \chi < \gamma \\ R(\frac{\chi}{\gamma}) * \gamma, & \text{if } \gamma \leq \chi < L \\ L, & \text{if } \chi \geq L \end{cases} \\
h &= \begin{cases} \gamma, & \text{if } \chi < \gamma \\ max(R(\frac{w*\lambda}{\gamma}) * \gamma, \gamma), & \text{if } \gamma \leq \chi < L \\ L, & \text{if } \chi \geq L \end{cases} \\
\chi &= max(W, H) \\
R(x) &= \lfloor x + \frac{1}{2} \rfloor \\
\lambda &= \frac{H}{W}
\end{aligned}
\tag{3}
$$

where H and W are the input image width and height, γ is a multiple of 32 and L is the hyperparameter ceiling with a standard value of 416.
The output of the RA-YOLO module is called PD info (pedestrian detection information). The image is then subsequently divided through *intelligent splits* (Figure 3) via a two-step procedure. Firstly the bounding boxes considered outliers are removed using the Z-score method, and then the image is divided into two sub-images based on the remaining bounding boxes. The Z-score method is a statistical technique used to standardize and quantify the deviation of an individual data point from the mean of a dataset. It involves subtracting the mean from the data point and then dividing the result by the standard deviation. The resulting Z-score indicates how many standard deviations a data point is from the mean, providing a measure of its relative position within the distribution. This study flags an element as an outlier if either its width or height z-score exceeds a certain threshold and its confidence score is below a chosen parameter. Each image is then transferred to RA-YOLO to obtain PD info with a higher detail. The last step consists of merging the various PD info obtained through a method similar to non-maximum suppression (NMS) [36]. The Non-Maximum Suppression (NMS) algorithm takes a set of bounding boxes with associated confidence scores as input, sorts them based on confidence in descending order, and outputs a list of selected bounding boxes after suppressing overlapping boxes using the IoU threshold. The pseudo-code for NMS is reported in Algorithm 1. This approach has been able to detect

Algorithm 1 Non-Maximum Suppression (NMS)

```
Input: Set of bounding boxes B and their associated confidence scores S
Output: List of selected bounding boxes after NMS
Sort B based on S in descending order
Create an empty list selected_boxes
for box in B do
    if box has not been suppressed by higher-scoring box in selected_boxes then
        Add box to selected_boxes
        Suppress overlapping boxes in B with box using IoU threshold
    end if
end for
return selected_boxes
```

many pedestrians that were not detected by other state-of-the-art models, obtaining an AP of 88.5% on the PASCAL VOC 2012 [11].

To improve the generalization capabilities of a model, Hasa et al. [18] focused on the data instead of on the model. They demonstrated how classical pedestrian detectors, which are tailored to specific pedestrian detection datasets, perform poorly in cross-dataset evaluation when compared to general object detectors trained with diversified and extensive datasets. To do so, the authors proposed a *progressive training pipeline*, that consists of starting from a general diverse dataset in terms of scenes and density and then training on a more specific dataset. The experiments have shown how this training method can substantially improve the generalization capabilities of Cascade R-CNN [5], a state-of-the-art model. For example it improved the performance on the Citypersons dataset [65] using progressive training with the Wider Pedestrian dataset and the Eurocity Persons dataset [3] it has obtained an MR^{-2} (log average miss rate over false positive per image with range from 10^{-2} to 10^{0}) of 9.7, while a score of 10.9 when concatenating those two datasets and the CrowdHuman dataset [46]. Attention was also given to small and light-weight models, since in many pedestrian detection applications the device on which the model operates imposes constraints on its size and computing cost. The model taken into consideration was MobileNet V2 [21], which is also used in this work, and the improvements hold even for such an architecture designed for mobile and embedded applications.

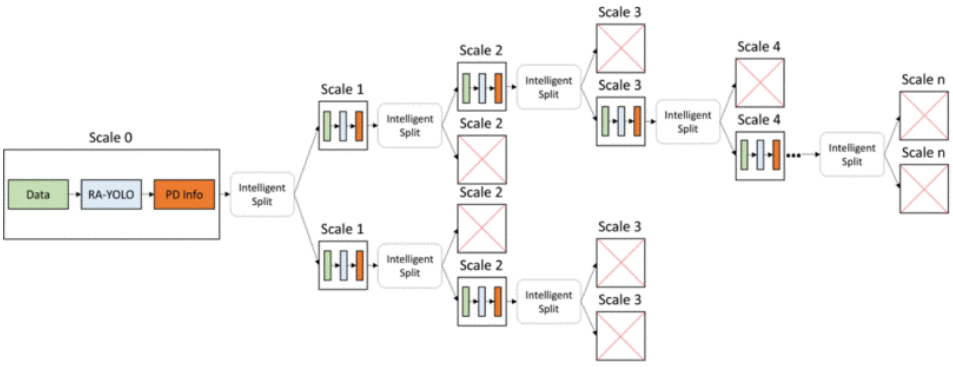


Figure 3: Representation of *intelligent splits*, where scale 0 corresponds to the original image and next scales are obtained via iterative splits [22]. The first step involves filtering out outliers of the pedestrian bounding boxes, and the second step involves separating the input images into two sub-images.

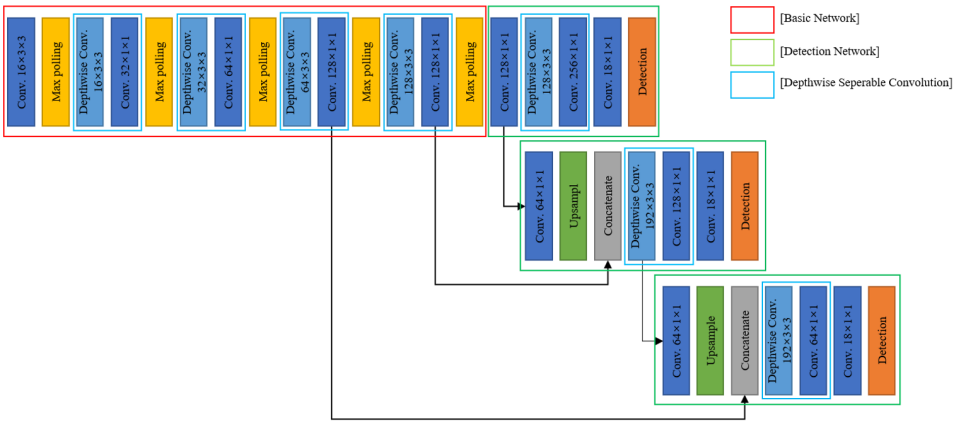


Figure 4: Overview of the proposed custom Tiny-YOLOv3 network architecture [24]. At the end of the classical architecture, two more branches are added that use depthwise convolution to improve the model accuracy.

Besides the detection of the body bounding box, some studies focused on head detection for various applications. The authors of [24] developed a system to perform automatic passenger counting and direction recognition applicable to a non-GPU embedded system. The model proposed is a modification of the Tiny-YOLOv3 network [43]. The backbone is simplified by removing and reducing layers and the number of convolutional filters and by removing the batch normalization layers, which slow down the inference time. At the end of the network, two branches are added to improve the detection accuracy. Having added two branches, to keep the model lightweight and fast depthwise convolution is adopted [21] and the size of the input images is reduced from 416 x 416 (the default for Tiny-YOLOv3) to 224 x 224. A detailed description of the architecture is shown in Figure 4. Experiments have shown how, despite being significantly lighter than the reference model (1.58 BFLOPS and 0.14 BFLOPS), it outperforms its accuracy on all the tested datasets besides being highly performing, running at 26.34 FPS with an AP of 14.23 on CrowdHuman against an AP of 9.47 at 4.14 FPS for the baseline model. In recent years a lot of studies have involved the use of the *attention mechanism* [37], for example in Transformers [55], especially in tasks like natural language processing and time series forecasting. In CFNet (cross-layer and feature fusion and fusion weight attention network) [17] the attention mechanism is used to give importance to different features. An overview of the framework is available in Figure 5. The network has two major modules: the cross-layer feature fusion and the fusion weight attention module (FWAM). The first combines the features of three different resolutions (104 x 104, 52 x 53, and 26 x 26) since different feature layers contain different information and their combination can enable better detection results. To avoid redundancy of feature information weights are needed for each level. The original features are first convoluted to obtain three different features p_1, p_2 and p_3 that are combined using three different weights w_0, w_1 and w_2. The features and

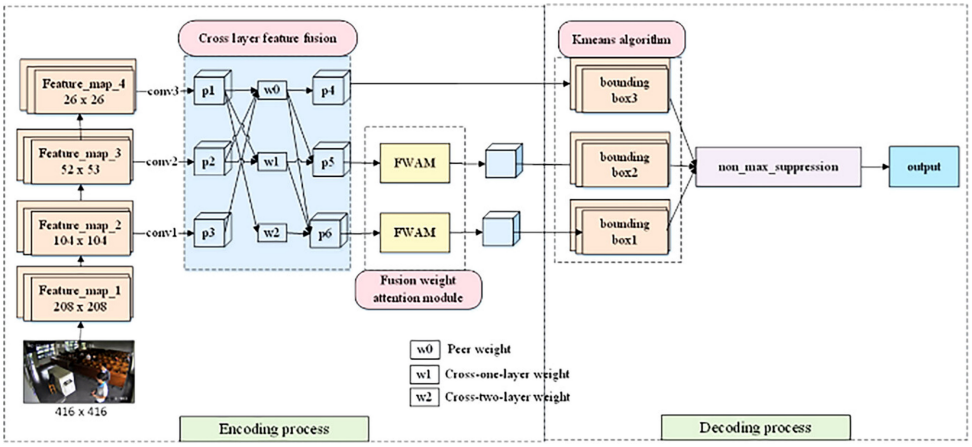


Figure 5: Complete CFNet pipeline [17]. It utilizes a cross-layer feature fusion module to combine features from different layers, optimizing inter-layer coefficients for optimal fusion and addressing the challenge of large feature gaps across layers. Additionally, a fusion weight attention module assigns varying weights to the fused feature information, considering both spatial and channel aspects.

weights are combined to obtain the output features p_4, p_5, and p_6 using the following equations:

$$\begin{aligned} p_4 &= p_1 \cdot w_0 \\ p_5 &= p_2 \cdot w_0 + p_1 \cdot w_1 \\ p_6 &= p_3 \cdot w_0 + p_2 \cdot w_1 + p_1 \cdot w_2 \end{aligned} \tag{4}$$

The two fused features with the highest resolution p_5 and p_6 are then fed to a FWAM, which through the attention mechanism assigns higher weights to features with strong information to optimize head detection. This operation is performed in the convolutional block attention module (CBAM) [61] divided into two parts: channel attention, and spatial attention. The two modules receive simultaneously the features and compute the respective weights, which are then combined to obtain the fusion weight matrix. Finally, this matrix is grouped with the input features, highlighting the important features. These highlighted features correspond to three groups, one for each different resolution: one is the direct output of the cross-layer feature fusion module, and the other two are from the FWAMs. From each of these groups, one bounding box is obtained via the k-means algorithm. Finally, they are filtered through NMS and the detection boxes with intersection over union (IOU) less than 0.5 are removed. This method has shown robustness and better performance than other networks at the cost of a greater computational cost, obtaining an AP of 94.22% at 22 FPS on the SCUT-HEAD PartA dataset [40].

In [52] a novel method for online multiple pedestrian tracking based on both head and body detection is proposed. The motivation is that the detector and most state-of-the-art detectors influence the tracking performance and rely on one single detection algorithm, while the detectors that use other classes (e.g. such as head detections or body key points) combine body and head detections offline. Another problem that this work tries to mitigate is that in the case of significant occlusions, which is often in crowded scenarios, detectors often fail while head occlusions are relatively small. The main operation is represented by the data association performed between head and body detections, which is carried out by exploiting the ratio between the head box and the body box dimensions to produce possible matches:

$$\begin{aligned} w_j^b &= C1 w_i^h, \\ h_j^b &= C2 h_i^h, \\ y_j^b &= y_i^h, \\ x_j^b &= Z x_i^h + \beta, \end{aligned} \tag{5}$$

where C1 and C2 are default parameters and Z and β are obtained from training a linear regressor to find the relationship between the head and body bounding boxes on the x-axis. Afterwards the Hungarian algorithm [25] computes the best pair of head and body minimizing a cost function composed of the IOU and the top border distance. Knowing the ratio between the head and body dimensions enables the model to be able to reconstruct the head bounding box when the body is not detected and vice-versa, enhancing the tracking performance. Experiments demonstrated how this method is superior to other trackers, with higher accuracy

and fewer detections missed and false detections, reaching up to 98.3% in precision in the MOT16 dataset [34] and 95.2% in the MOT20 dataset [9].

2.2. Age Estimation

The age estimation problem can be interpreted both as a regression and classification task (as introduced in Section 1), Sharma et al. [47] proposed an improved CNN to perform age classification besides gender estimation. The structure of the network is the standard one for a CNN: the feature extraction section and the classification one. The feature extractor is composed of four convolutional blocks, each one composed of a convolutional layer, a Relu function, and a max pooling layer. The first and second blocks use convolutional layers with 64 x 3 x 3 filters and the third and fourth ones with 128 x 3 x 3. After a max-pooling layer with a 2 x 2 window and a flattening layer, the classifier receives the extracted features. After two fully connected layers of 128 and 98 nodes and a softmax activation function, the probability distribution among the 98 classes of age (one class for each age) is given as output. The input images are resized to 150 x 150. Despite the simple structure, after being trained and tested on the UTKFace [66] it has obtained a test MAE (mean average error) of 0.77 and on cross-evaluation a MAE of 2.9 on IMDB-WIKI [45], 3.9 on FG-NET and 7.4 on CACD [6].

Similarly to the approach chosen in [18], Zhang and Bao [64] introduced an end-to-end learning model called cross-dataset training CNN (CDCNN), treating age estimation as a classification problem. The key point in this work is to merge different datasets, that usually contain images from a single ethnicity, to create a training set that can improve the generalization capabilities of the model. As a preprocessing step, all the facial images are resized to 256 x 256 and then randomly cropped to 224 x 224 pixels and then a CNN is used to classify the images. The architecture chosen is VGG-16 [51] pretrained on ImageNet, with the last layer modified from 1,000 outputs to the number of desired age classes, in this work from 14 to 71 years of age. This choice has been driven by the fact that VGG-16 has obtained very good results on different challenges, besides the fact that pretrained models are available. The results of the experiments show how cross-dataset can improve the model performance, lowering the MAE on the CACD test set from 4.58 without cross-dataset training to 3.96 when trained using also the MORPH dataset [44], while improving from 3.30 to 3.11 on the AFAD test set [38], always with MORPH in cross-dataset training.

Besides choosing to solve age estimation as a regression or classification problem, Shin et al. [50] proposed to tackle the problem as an ordinal regression problem, introducing the Moving Window Regression (MWR) algorithm. Ordinal regression aims to predict an instance's position within a list of references based on age as the ranking metric. Rather than directly forecasting a subject's age, it estimates their relative rank compared to other references. This process is iterated to improve the prediction. The essential concept for MWR is *ρ-rank*

$$\begin{aligned} \rho(x, y_1, y_2) &= \frac{\theta(x) - \mu(y_1, y_2)}{\tau(y_1, y_2)} \\ \mu(y_1, y_2) &= \frac{1}{2}(\theta(y_1) + \theta(y_2)) \\ \tau(y_1, y_2) &= \frac{1}{2}(\theta(y_1) - \theta(y_2)) \end{aligned} \tag{6}$$

where x is the input, y_1 and y_2 are two references with $\theta(y_1) < \theta(y_2)$, where θ returns the rank of the given input. Thus the aim is to predict the ρ-rank of the input instead of the absolute rank θ. One the ρ-rank is predicted, the absolute rank θ can be reconstructed:

$$\theta(x) = \rho(x, y_1, y_2) \cdot \tau(y_1, y_2) + \mu(y_1, y_2) \tag{7}$$

The ρ-rank has the following properties: $\rho \in [-1, 1]$ and it represents how close the given value is to one of the references, meaning that if the value is positive it is closer to the higher reference and vice versa. The absolute value of ρ quantifies how much it is close to the references y_1 or y_2 , with $|\rho| = 1$ when $|\rho| = y_1$ or $|\rho| = y_2$.
To estimate the ρ-rank a *ρ-regressor* has been developed, composed of an encoder and a regression module. The encoder module adopted is VGG-16, which takes as input the triplet (x, y_1, y_2) and extracts at the same time the features $f(x)$, $f(y_1)$, and $f(y_2)$. These features are then fed to the regression module, made of three fully-connected layers, that estimate ρ. To facilitate the learning process the rank difference between input references is constrained to be a fixed value, so to have a smaller subset of (x, y_1, y_2) to train on.
The inference procedure is structured as follows. An initial estimate is obtained by extracting the input image features and then estimating its ρ-rank averaging and rounding the ranks obtained via K NNs in the feature space. The metric used is the Euclidean distance and K is set to 5. After the first estimation $\hat{\theta}^0(x)$ is obtained, it is iteratively refined using reference values y_1^1 and y_2^1 such that

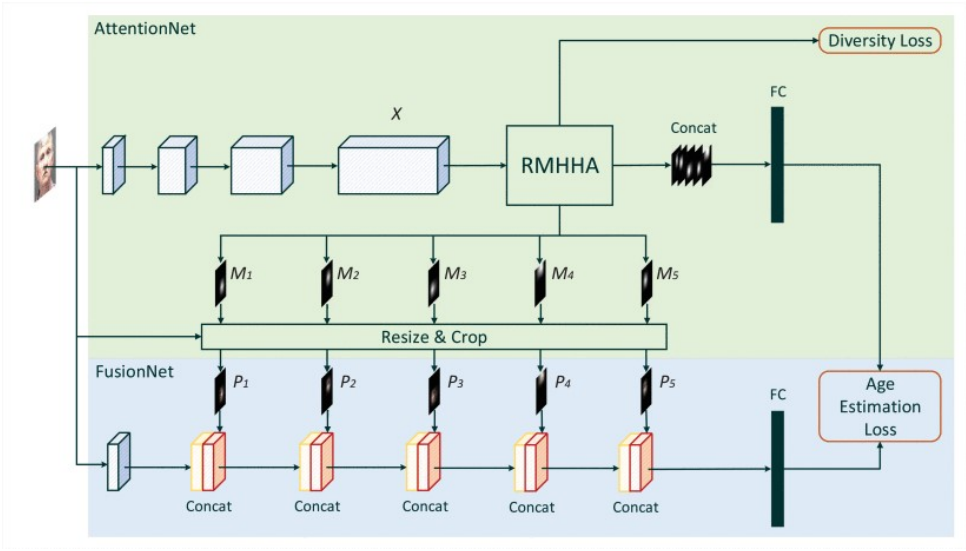


Figure 6: ADPF structure [59]. The system comprises two networks, AttentionNet and FusionNet. AttentionNet trains a proposed mechanism, RMHHA, to learn and rank age-specific features denoted as M1 to M5. Once ranked, these features are resized and cropped from the input facial image, forming patches P1 to P5 based on the amount of age-specific information. The network architecture includes CNN layers, concatenation operations (Concat), and fully-connected layers (FC). The color-coded blocks represent layers from the main stream (yellow) and age-specific patches (red), while X denotes the input tensor to the RMHHA mechanism with dimensions $32 \times 32 \times 500$.

$$\begin{aligned} \theta(y_1^1) &= \hat{\theta}^0(x) - \tau \\ \theta(y_2^1) &= \hat{\theta}^0(x) + \tau \end{aligned} \tag{8}$$

y_1^1 and y_2^1 represent the extremes of the *search window*, among which the value of new rank estimation is computed. These operations are repeated until the new estimate is equal to the new one or if the maximum number of iterations is reached. The reference pairs to be considered are selected offline, minimizing the regression error γ (Equation (9)).

$$\begin{aligned} \gamma(y_1, y_2) &= \frac{1}{|W|} \sum_{x \in W} |\hat{\rho}(x, y_1, y_2) - \rho(x, y_1, y_2)| \\ W &= x | \theta(x) \in |\theta(y_1) - \alpha, \theta(y_1) + \alpha| \end{aligned} \tag{9}$$

To improve the regression accuracy, two different kinds of regressors are trained: global and local ρ-regressors. A global regressor is used on the entire age range, while local ones on the specific ranges. Since the characteristics of different age groups show large differences, local regressors can be tailored to specific ranges to improve accuracy. The full age range is divided into 5 overlapping subsets, each one assigned to a local regressor, where each group shares the first half with the previous group and the second half with the following group (except for the first and last group that only share the second and first half respectively). At inference time, the first estimation is performed with the global regressor, and the next ones with the local regressors based on the age group to which the estimate belongs. Due to the overlaps, the value can belong to two age groups, in that case, two local regressors are used and their result averaged. The proposed method obtained a MAE of 2.23 on FG-NET, 4.37 on UTKFace, and 5.68 on CACD.

In recent years research interest in the attention mechanism has grown due to its ability to assign varying importance to different features, in contrast to CNNs, which treat all features equally. Following this trend, Wang et al. [59] implemented Attention-based Dynamic Patch Fusion (ADPF). The proposed architecture is composed of two distinct convolutional neural networks (CNNs): AttentionNet and FusionNet, depicted in Figure 6. AttentionNet uses Ranking-guided Multi-Head Hybrid Attention (RMHHA) to identify and prioritize age-specific features and weigh them accordingly. FusionNet then combines these weighted features, along with the ones extracted from the input image, to generate an age estimation. The RMHHA module is based on the Multi-Head Self-Attention (MHSA) [55], wherein each head the input tensor (with size h x w x c_i) is sent through a 1 x 1 convolutional layer to obtain three tensors Q, K and V, with the same dimensions h x w

and different number of channels c_Q, c_K and c_V, with $c_Q = c_K$. These tensors are then combined to produce *self-attention maps* (SA) with dimensions h x w x c_V:

$$SA = Softmax(\frac{Q' \cdot K'^T}{\sqrt{c_K}}) \tag{10}$$

where Q' and T' are the flattened tensors.
The SA obtained is a flattened tensor, and because the problem involves images, it is converted into matrix form the ensure efficiency. The input tensor is also passed through another 1 x 1 convolutional layer to obtain a tensor CA with size h x w x c_V, that after two fully connected layers is transformed into the *Channel-wise Attention Weights* (CAW). The last operation is a weighted summation between the SA and CAW to obtain the final *Hybrid Attention Map* (HA):

$$HA = \sum_{c}^{cV} SA_c \cdot w_{CA} \tag{11}$$

where c is the channel and w_{CA} are the CAW. The model deploys five different attention heads, and their resulting HAs are weighted via learnable weights, before being concatenated. To avoid the overlapping of patches a special diversity loss is minimized. Therefore the output of the AttentionNet module is represented by the final concatenated HA and the HAs of the different heads. These HAs are ranked based on their weights and then used on the input image to crop the highlighted areas, therefore creating five different patches. The process involves merging the patches by concatenating them in pairs of feature maps. This results in the final output R.:

$$R = Concat[I,\ P] \tag{12}$$

where I represents the features coming from the previous layer, and P is the current patch being merged. To estimate the age the features are processed by a Softmax function, the negative values are eliminated and the remaining values normalize to have a sum equal to 1, and the final prediction is computed as:

$$E = \sum_{p=1}^{q} o_p g_p \tag{13}$$

where o_p is the probability of class p and g_p the class label p.
The downside of this architecture is that the inference time required is higher than other works due to the hybrid attention mechanism, but it has scored a MAE of 2.56 on FG-NET and 5.39 on CACD.

2.3. Gender Classification

As mentioned in the paper by Sharma et al. [47], a CNN model was proposed for gender classification from face images. The architecture of the network is similar to the one used for age classification. However, it has been simplified and includes only two convolutional blocks with 64 x 3 x 3 filters, a 2 x 2 max-pooling layer, a flattening layer, and two fully-connected layers of 64 and 1 node. The output is passed through the sigmoid activation function. The experiments carried out have shown an excellent accuracy of 99.86% on the UTKFace dataset.

Sheikh and Heidari [48] utilized the central difference convolution (CDC) [63] to modify a typical CNN network applied to facial images. The approach involves using two parallel CNNs: one with the traditional convolution method, known as the vanilla convolutional network (VCCN), and the other with the newly introduced CDC, referred to as the central difference convolutional network (CDCN).
While the output feature map of a vanilla convolution for a location p_0 can be defined as:

$$y(p_0) = \sum_{p_n \in R} w(p_n) \cdot x(p_0 + pn) \tag{14}$$

the central difference convolution incentivizes centre-oriented gradient of sampled values, therefore Equation (14) becomes:

$$y(p_0) = \sum_{p_n \in R} w(p_n) \cdot (x(p_0 + pn) - x(p_0)) \tag{15}$$

To take advantage of both convolutions is possible to combine Equation (14) and Equation (15) to generalize central difference convolution:

$$y(p_0) = \theta \cdot \sum_{p_n \in R} w(p_n) \cdot (x(p_0 + pn) - x(p_0)) + (1 - \theta) \cdot \sum_{p_n \in R} w(p_n) \cdot x(p_0 + pn) \tag{16}$$

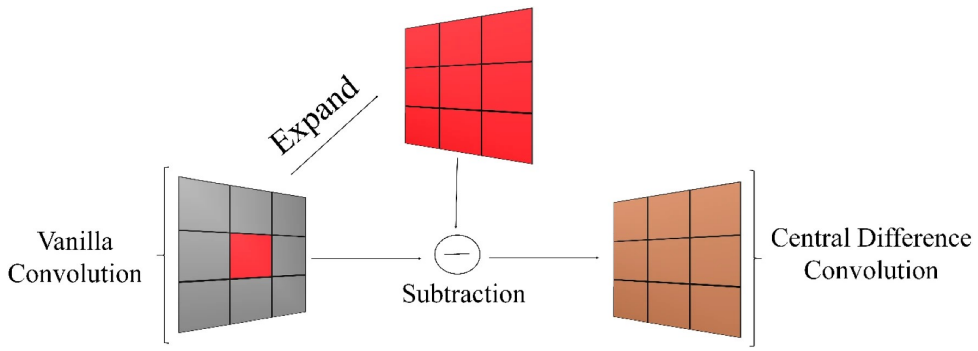


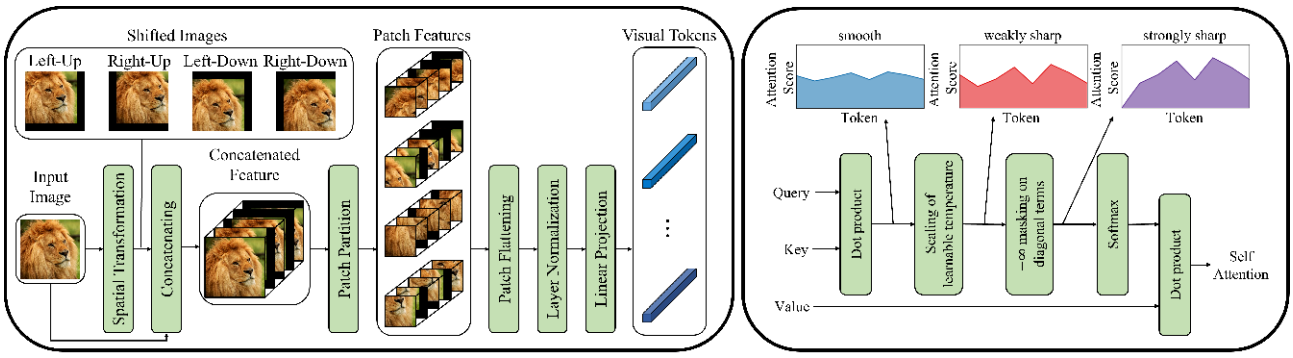
Figure 7: A visual representation of central difference convolution [48].

with $\theta \in [0, 1]$ interpolates between the intensity-level (vanilla convolution) and gradient-level information (central difference convolution). A visual overview of central different convolution is depicted in Figure 7. Before being fed to the two parallel networks VCCN and CDCN, the input image is resized to 128 x 128, and then the output features of the two networks are concatenated and fed to the classifier to predict the input gender. VCCN is composed of three blocks, each one made of three vanilla convolution (VC) layers and one max-pooling layer. The VC layers have sizes 32 x 3 x 3, 64 x 3 x 3, and 128 x 3 x 3 in the first, second and third blocks respectively. In contrast, the pooling window is 2 x 2 in the first block and 4 x 4 in the following ones. The classifier has three fully-connected layers of 150, 100, and 50 nodes, and the 2 output nodes, representing the male and female classes. This method has obtained an accuracy of 99.1% on the FEI dataset [4] and 97.79% on the LWF dataset [23].

Fayyaz et al. [12] proposed an architecture to combine traditional hand-crafted features and CNNs to perform gender classification from full-body images. The traditional features employed were the HOG (Histogram of Oriented Gradients) [8] and LOMO [27] descriptors, while VGG19 [51] and ResNet101 [19] as CNN feature extractors.
The HOG descriptor extracts shape information and the LOMO descriptor embodies information about colour and texture, therefore combined can reliably represent an image. The HOG descriptor about illumination and rotation invariant features of an image is computed by dividing it into different blocks to get the image orientation, and then the gradient of the image is computed in blocks of size 2 x 2. From every block, an orientation histogram of nine beans is computed and normalized using L1-norm with interpolation. All the local histograms are finally concatenated to obtain a global histogram, which is still normalized using the L2-norm. The resulting HOG feature vector is 3780. LOMO features are extracted using a moving window with a size of 10 x 10 and stride of 5, with 128 x 64 patches. Each sub-window comprises two types of information: scale-invariant local ternary patterns (SILTP) [28] and HSV-based 8 x 8 x 8 colour bin histograms. A three-level pyramid is utilized to extract information at multiple scales. The information from different windows and scales is then combined, resulting in a feature vector with a size of 26,960.
Besides the low-level feature vectors, pre-trained ResNet101 and VGG19 extract high-level features, resulting in vectors of size 1000 and 4096 respectively. Before fusing all the feature vectors obtained from HOG, LOMO, VGG19, and ResNet, a feature selection process is carried out to reduce the feature size and to keep only the most relevant ones. This process is based on maximizing the entropy and after extensive experimentation, the maximum number of features kept from HOG, ResNet101, and VGG19 is 1000, while 600 for LOMO, for a total size of 3600.
Different classification methods are employed to train the classifier: discriminant analysis, ensemble, KNN (K Nearest Neighbours), and SVM (Support Vector Machine). The best results are achieved with the cubic SVM scoring an accuracy of 89.3% on the PETA dataset [10].

Authors in [1] propose ViT-PGC, a framework for pedestrian gender classification using ViT to overcome the limited receptive field of CNNs.
The first step in the pipeline is to preprocess the input image performing contrast adjustment and applying a median filter to remove the noise. The transformer is based on swin architecture [32], with the addition of two modules: the shifted patch tokenization (SPT) and the locality self-attention (LSA) modules. They both mitigate the lack of locality inductive bias and enhance learning from scratch even on small-sized datasets. SPT spatially translates each image in one of four directions (down-left, up-left, down-left, down-right), then the images are concatenated and split into non-overlapping patches to obtain, that are then flattened, where each component is computed as:

$$P(x) = [x_p^1; x_p^2; ...x_p^N] \tag{17}$$



(a) Shifted Patch Tokenization (b) Locality Self-Attention

Figure 8: An architecture overview of the SPT and LSA modules [26]. The Spatial Patch Transformation (SPT) module improves Vision Transformers (ViTs) by spatially shifting input images, merging them into a single image, and applying patch partitioning. Subsequent operations include layer normalization, patch flattening, and linear projection. The LSA (Learnable Self-Attention) module serves as a regularization function, adjusting attention score distributions by learning temperature parameters in self-attention mechanisms. It incorporates a diagonal mask in the generation of the similarity matrix for query and key, eliminating self-token relations and increasing attention scores for different tokens.

where $x_p^i \in R^{P^2 \cdot C}$ is the i^{th} flattened vector, P is the patch size and and $N = \frac{HW}{p}$ the number of patches. After this a linear projection and normalization are applied and the tokens are obtained. The main upside of SPT is that it can encode additional spatial information and improve ViT's locality inductive bias. A typical ViT then learns the linear projection applied to each token for query, key, and value. The next operation is to compute the similarity matrix and the attention score matrix is produced with the softmax function:

$$SA(x) = Softmax(\frac{R}{\sqrt{d_k}})xE_v \tag{18}$$

where R is the similarity matrix, E_v the linear projection of value and d_k the dimension of the key. The LSA module is a general regularization function to control and sharpen the distribution of the attention scores by learning the temperature parameters of the softmax function in the self-attention mechanism. It also introduces a diagonal mask when producing the similarity matrix of query and key, removing the self-token relation, and raising the attention score of different tokens. A review of SPT and LSA can be found in Figure 8. This architecture has obtained an accuracy of 91.7% on the MIT dataset, but a decrease in the accuracy was detected on cross-dataset evaluation.

3. The PAGE Framework

In this section, we introduce the PAGE framework, explaining its components and the motivations behind our design choices. An overview of the pipeline is available in Figure 9. In the last part of the section, we describe the experiments performed and their results.
The pipeline consists of three main modules:

1. YOLOX-nano, responsible for detecting both heads and bodies;
2. Head Analysis, which receives a head image and estimates the age and gender of the subject;
3. Body Analysis estimates the gender from the full body image.

YOLOX first processes the input image to extract the detections, which are subsequently filtered and used to obtain body and head images. The sub-images are then preprocessed and batched before being analysed by the Head and Body Analysis modules. The resulting labels are finally added to the detected bounding boxes and head-body association is performed. The resulting image has both detection boxes with the respective gender and age labels.

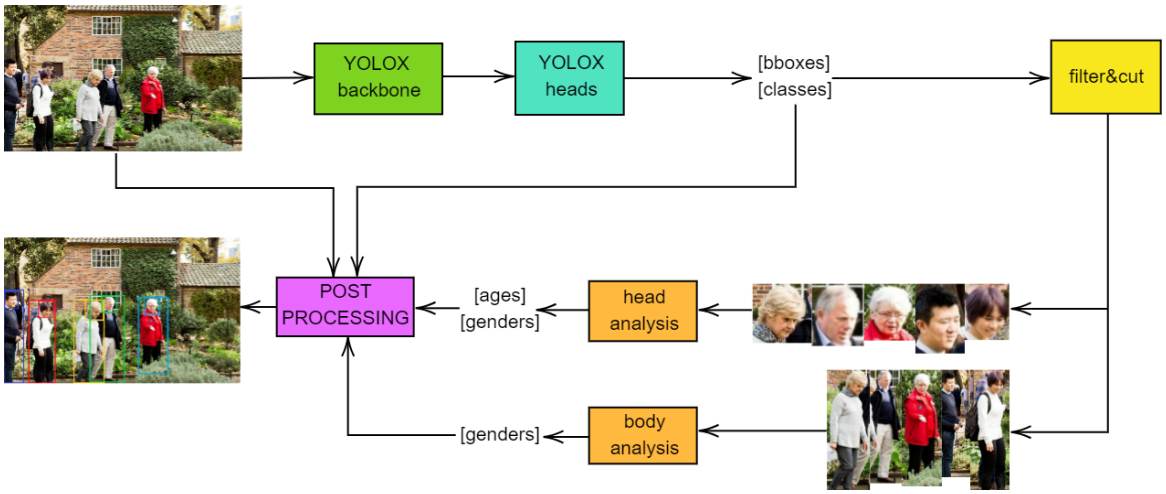


Figure 9: An overview of the framework. The input image is first processed by YOLOX, then the detections are filtered and head and body images are obtained. The sub-images are then analysed by the respective modules to obtain the age and gender labels. The final step is post-processing, where the input image, the detection boxes and the labels are fused to obtain the output image.

3.1. YOLOX-nano

YOLOX [14], already introduced in Section 2, has been selected among the detectors available due to its good balance between accuracy and speed, besides the fact that full code and extensive documentation are made available by the authors. Moreover, the YOLOX-nano version has been considered because it has been specially designed to run on mobile devices.
An overview of YOLOX architecture is available in Figure 10. The backbone of the network is CSPDarknet53 [58], where features at three different layers are extracted and sent to the neck of the network. The levels selected are obtained at the 82^{nd}, 94^{th} and 106^{th} layers respectively, which result in outputs of sizes 13 x 13, 26 x 26 and 52 x 52. Thus, the three different outputs are used to detect objects of big, medium and small sizes.
After the feature extraction network, the three-level features are processed by the neck and then by three different heads. The neck comprises a Path Aggregation Network (PAN) [30] and a FPN. These two networks are responsible for combining the obtained features in order to exploit both high-level and low-level semantic information. At the end of the network each head, detailed in Figure 11, is responsible for finding the bounding boxes and the respective object class. Since it is common to detect more bounding boxes for the same object, NMS is performed to keep only the best box for the object. Before being processed by the Head and Body Analysis modules the detected boxes are filtered, removing the ones that do not belong to the person or head class. After that, the detected box values (centre coordinates and dimensions) are used to cut the original input image to isolate the body or head image. Finally, the head and body images are preprocessed and batched to speed up the computation.
The preprocessing functions are necessary to resize the images to the expected size by the Head and Body Analysis modules, where for head images the expected size is 80 x 50 and 250 x 100 for the body images. First,

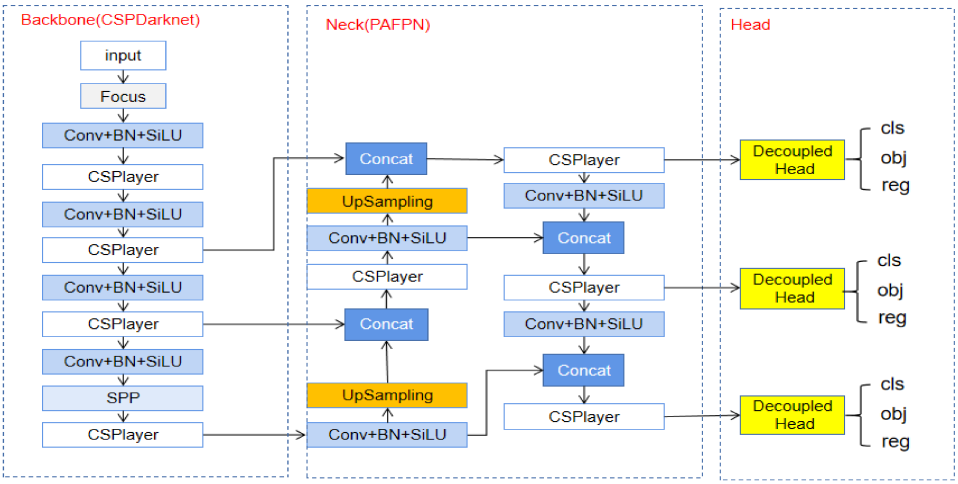


Figure 10: Full YOLOX structure [62]. YOLOX utilizes the YOLOv5 Focus module for channel optimization and the CSPDarkNet feature extraction network from YOLOv4 as its backbone. The YOLOX neck uses the PAFPN structure to blend multi-scale features seamlessly. Following processing through the backbone and neck networks, the input image is segmented into three scales (13×13, 26×26, and 52×52) for predicting large, medium, and small objects. The next step is to feed the features from these scales into the decoupled detection head for efficient object detection.

Algorithm 2 Image Preprocessing

Input: image # Original image to be processed
target_width, target_height # Target dimensions for width and height
Output: padded_image # Resized and padded image
original_width ← width of image
original_height ← height of image
if original_width > target_width **then**
    new_width ← target_width
    new_height ← target_width $\times \left(\frac{\text{original\_height}}{\text{original\_width}}\right)$
else
    new_width ← original_width
    new_height ← original_height
end if
if new_height > target_height **then**
    new_height ← target_height
    new_width ← target_height $\times \left(\frac{\text{original\_width}}{\text{original\_height}}\right)$
end if
resized_image ← resize image to (new_width, new_height)
padded_image ← pad resized_image to (target_width, target_height)
return padded_image

an image is resized keeping the original aspect ratio until one of the dimensions reaches one of the desired sizes, and then the rest of the image is padded to adjust the other dimension. The pseudo-code for this operation is reported in Algorithm 2, while examples of original and preprocessed images are given in Figure 12.

3.2. Head Analysis

The Head Analysis module is responsible for taking as input a head image and estimating the gender and age of the subject. The main idea is to exploit the features extracted by CSPDarknet (YOLOX backbone) to perform the predictions. Two different networks are employed: one for age regression and one for gender classification. Once the labels are obtained they are sent to the post-processing stage. The loss used for age regression is the L1 loss, which measures the MAE between the output of the network and the target values:

$$\begin{aligned} L1(x, y) &= mean(L) \\ L &= \{l_1, ..., l_n\}^T \\ l_n &= |x_n - y_n| \end{aligned} \tag{19}$$

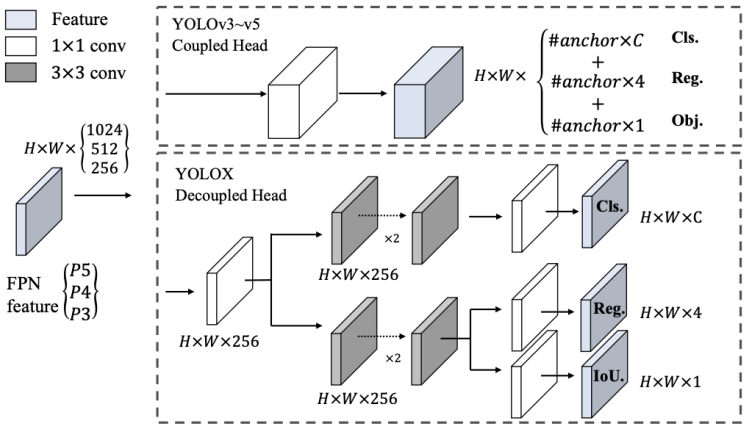


Figure 11: Comparison between baseline YOLO head and YOLOX decoupled head [14]. At each FPN level, a 1x1 convolution layer decreases the channel to 256. Two parallel branches follow, each with two 3x3 convolutional layers for classification and regression. An IoU branch is also added to the regression branch.

(a) (b) (c) (d)

Figure 12: Examples of preprocessing results. The original images (a) and (c) are processed to obtain (b) and (d). Images are scaled for clarity.

For gender classification, the chosen loss is the BCE with logits loss, which combines a *Sigmoid* function with the Binary Cross-Entropy (BCE) loss. This loss is preferred because it is more numerically stable than using the function and the loss separately.

$$\begin{aligned} BCE(x, y) &= mean(L) \\ L &= \{l_1, ..., l_n\}^T \\ l_n &= -w_n[y_n \cdot log\sigma(x_n) + (1 - y_n) \cdot log(1 - \sigma(x_n))] \end{aligned} \tag{20}$$

3.3. Body Analysis

The Body Analysis module is similar to the Head Analysis one, but it only estimates the gender given the full body picture. It uses the same backbone and then a network tailored to perform gender classification. The estimated gender label is then post-processed with the labels generated by the Head Analysis module, the detection boxes and the input image. The training loss is always the BCE with logits, previously introduced.

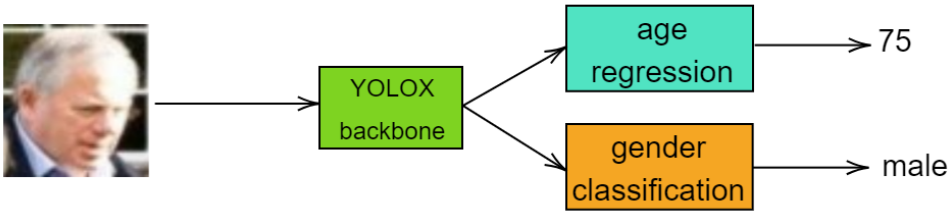


Figure 13: Overview of the Head Analysis module. The head image is firstly processed by the YOLOX backbone, then the features are passed to different models that predict the age and gender labels.

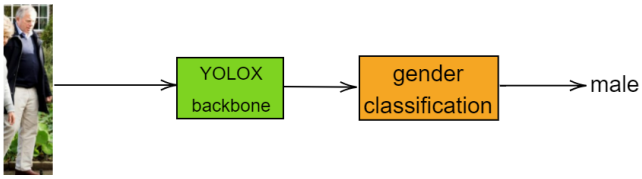


Figure 14: Scheme of the Body Analysis module. After being processed by the backbone the features extracted are fed to the classifier to estimate the gender label.

3.4. Postprocessing

After obtaining the age and gender from the head picture and gender from the body image, the bounding boxes need to be associated to link the head and body. This problem is solved by analysing the head boxes and for each of them computing the Interception over Head (IoH) with the body boxes, and then associating it with the one with the maximum score. The IoH is obtained with the following formula:

$$IoH(h, b) = \frac{A(h \cap b)}{A(h)} \tag{21}$$

where h and b are the head and body bounding boxes, $h \cap b$ their intersection and the function $A(x)$ computes the area of the given box. It follows that $IoH \in [0, 1]$, where $IoH = 0$ when there is no intersection and $IoH = 1$ when the head box is fully included in the body box. The boxes association method is reported in Algorithm 3. The final result is composed of different bounding boxes, each with the label class, the gender class, and the age class if it belongs to the head class.

3.5. Experiments

In this section, we describe the training procedure and the experiments performed, while in the next section, we discuss the results obtained.

YOLOX

Since YOLOX is trained on the COCO dataset, the available pre-trained models can detect objects belonging to 80 different classes, but among which there is not one representing a subject's head. To overcome this problem, YOLOX has been fine-tuned on the CrowdHuman Dataset [46] for 300 epochs, where its annotations have been converted to the COCO format in order to be compatible with the training procedure available. To detect only the body and head classes, the shape of the heads has been changed from 80 to 2 accordingly, while the input image size has been kept to 416 x 416 as standard.

Head Analysis

The datasets used to train the gender classification model from facial images are AFAD (Asian Face Age Dataset) [38], AgeDB [35] and UTKFace [66]. The model employed uses the first level of features extracted by the YOLOX backbone, then two convolutional blocks (Table 1) and finally a classifier (Table 2). The convolutional blocks have been added to allow the model to learn new features since the backbone has been trained on object detection and not classification, and it is meant to be used without being fine-tuned. To perform age regression the dataset chosen was the one obtained keeping a maximum of 1500 images for each age

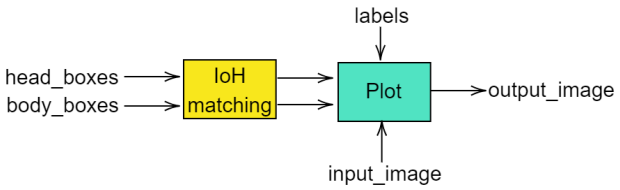


Figure 15: Post-processing procedure. The head and body bounding boxes are associated using IoH, then they are plotted with the age and gender labels on the input image to obtain the final image.

Algorithm 3 Boxes Association

Inputs: head_boxes # List of head boxes
body_boxes # List of body boxes
Output: max_pairs # Set of pairs with max IoH
max_pairs $\leftarrow \emptyset$
for $i \leftarrow 1$ to length(head_boxes) **do**
  max_IoH $\leftarrow 0$
  max_pair $\leftarrow \emptyset$
  for $j \leftarrow 1$ to length(body_boxes) **do**
    IoH $\leftarrow$ ComputeIoH(head_boxes[i], body_boxes[j])
    if IoH $>$ max_IoH **then**
      max_IoH $\leftarrow$ IoH
      max_pair $\leftarrow (i, j)$
    end if
  end for
  max_pairs $\leftarrow$ max_pairs $\cup$ {max_pair}
end for
return max_pairs

value. This dataset is later described in Section 4. Different models have been tested, the first is similar to the one used for gender classification that exploits some of the features extracted by CSPDarknet53 (Table 3 and Table 4). The other two models have ResNet50 (Table 5) and MobileNetV2 (Table 6) as backbones respectively. The data augmentation used is described in Section 4.

Body Analysis

The datasets used to train the gender classification model from full-body images are PA-100K [31] and PETA [10]. Following the same approach used for age regression from facial images, two different models have been tested to perform gender regression from full-body images. The first model always exploits the backbone of YOLOX-nano, followed by convolutional blocks and then a classifier (Table 7 and Table 8), while the second is a custom CNN (Table 9 and Table 10).

Operating Frequency

The objective of this experiment is to determine the actual performance of the pipeline in a real-time scenario. To have a better understanding of the weight of each different operation, we decided to report the different modules and their respective time. The experiment has been carried out in a lab environment using a docker container. The hardware used is composed of an NVIDIA GeForce GTX 1080 Ti GPU with an Intel Xeon E5-2630 v4 CPU.

3.6. Results

YOLOX

After the fine-tuning process, the model obtained an average precision (AP) of 21 for body boxes and 17 for head boxes, whereas the YOLOX-nano pretrained model had an mAP (mean Average Precision) of 25.8. Examples of detections are available in Figure 16. From these examples, it is possible to note how in some settings where the subjects are not too many and quite close to the camera, like Figure 16a and Figure 16c, the detections are pretty accurate. However, where the subjects are at a higher distance (Figure 16b) or there is a crowd (Figure 16d) the detector can fail.

Convolutional Blocks

Layer	Input Size	Output Size	Kernel Shape
Convolution	(64, 10, 7)	(256, 9, 6)	(2, 2)
ReLU	(256, 9, 6)	(256, 9, 6)	-
Max Pooling	(256, 9, 6)	(256, 4, 3)	2
Convolution	(256, 4, 3)	(512, 3, 2)	(2, 2)
ReLU	(512, 3, 2)	(512, 3, 2)	-
Max Pooling	(512, 3, 2)	(512, 1, 1)	2

Table 1: Convolutional part of the gender classifier from facial images.

Classifier

Layer	Size
Dense Layer	512
Dense Layer	512
Dense Layer	256
Dense Layer	256
Output	1

Table 2: Classifier structure for gender from facial images.

Convolutional Blocks

Layer	Input Size	Output Size	Kernel Shape
Convolution	(64, 10, 7)	(512, 9, 6)	(2, 2)
ReLU	(512, 9, 6)	(512, 9, 6)	-
Max Pooling	(512, 9, 6)	(512, 4, 3)	2
Convolution	(512, 4, 3)	(1024, 3, 2)	(2, 2)
ReLU	(1024, 3, 2)	(1024, 3, 2)	-
Max Pooling	(1024, 3, 2)	(1024, 1, 1)	2
Convolution	(1024, 1, 1)	(2048, 1, 1)	(1, 1)
ReLU	(2048, 1, 1)	(2048, 1, 1)	-

Table 3: Convolutional part of the age regressor from facial images.

Classifier

Layer	Size
Dense Layer	512
Dense Layer	512
Dense Layer	512
Output	1

Table 4: Classifier structure for age regression from facial images.

Head Analysis

The gender regression model obtained an accuracy of 92.56% on the test set. For age estimation, the model with CSPDarknet as its backbone has obtained a MAE of 7.89, while the others with different backbones are 7.46 when using ResNet50 and 8.17 MobileNet V2. While a good performance is obtained for the age estimation task, accurately predicting the age is harder. This can be attributed to the limited size of the models, combined with the data augmentation and the low resolution of the images.

Body Analysis

The models obtained an accuracy of 72.63% and 72.87% respectively when estimating the gender from full-body images. This decrease in the accuracy of the gender classifier from head images can be explained by the fact that when predicting the gender of an individual the most characteristic part is represented by the head. At the same time, the rest of the body is more similar. This phenomenon causes most of the features to be non-discriminant.

Operating Frequency

The results over ten runs and the average times are reported in Table 11. The columns are divided into:

- Detection: the YOLOX model that performs detection of heads and bodies;
- Association: that tries to couple each head with the respective body;
- Filter&Cut: head and body images are obtained from the input image thanks to YOLOX detections;
- Head and Body Analysis: estimate age and gender from facial and full body image;
- Other: all the other operations, which include plotting the bounding boxes and labels on the final image.

ResNet50 Classifier

Layer	Size
Dense Layer	512
Dense Layer	512
Dense Layer	512
Output	1

Table 5: Classifier structure with ResNet50 backbone.

MobileNet V2 Classifier

Layer	Size
Dense Layer	1024
Dropout Layer 0.5	-
Dense Layer	512
Dropout Layer 0.5	-
Dense Layer	256
Dropout Layer 0.5	-
Output	1

Table 6: Classifier structure with MobileNet V2 backbone.

Convolutional Blocks

Layer	Input Size	Output Size	Kernel Shape
Convolution	(64, 32, 13)	(256, 31, 12)	(2, 2)
ReLU	(256, 31, 12)	(256, 31, 12)	-
Max Pooling	(256, 31, 12)	(256, 15, 6)	2
Convolution	(256, 15, 6)	(512, 14, 5)	(2, 2)
ReLU	(512, 14, 5)	(512, 14, 5)	-
Max Pooling	(512, 14, 5)	(512, 7, 2)	2
Convolution	(512, 7, 2)	(1024, 6, 1)	(2, 2)
ReLU	(1024, 6, 1)	(1024, 1, 1)	-

Table 7: Convolutional part of the gender classifier from full-body images.

Classifier

Layer	Size
Dense Layer	512
Dropout Layer 0.5	-
Dense Layer	256
Dropout Layer 0.5	-
Dense Layer	128
Dropout Layer 0.5	-
Output	1

Table 8: Classifier structure for gender classification from full-body images.

To better represent how much time each phase takes, the results have also been expressed in percentage in Figure 17. It is possible to note how on average the most expensive phase is the detection one, followed by the other operations, the head analysis, the body analysis and the filtering and cutting task. The least expensive operation is the box association. As expected, the head analysis module takes more time than the body analysis one, having to compute both gender and age labels, but not double the time since body images have higher resolution (80 x 50 and 250 x 100). From the total average time, we can obtain the working frequency:

$$\text{FPS} = \frac{1}{time} = \frac{1}{0.4287} = 2.334 \text{ FPS} \tag{22}$$

The resulting operating frequency can be considered adequate when dealing with walking pedestrians, also considering that the output is obtained not via one single model but by combining four different ones and performing operations on images that are notoriously onerous. Nonetheless, further experiments on different hardware settings can be performed to obtain more accurate information about the operating frequency and how much the operations influence the pipeline.
To assess the efficacy of integrating pedestrian detection and analysis into a single framework as opposed to employing cascaded autonomous models, we can evaluate the total operating frequencies of the independent models and compare them with those measured for PAGE. To compare these values we use the ones reported from each author, therefore using different hardware settings. Carrying out a deeper comparison with the same setting is left for future research.

Convolutional Blocks

Layer	Input Size	Output Size	Kernel Shape
Convolution	(3, 250, 100)	(64, 248, 98)	(3, 3)
ReLU	(64, 248, 98)	(64, 248, 98)	-
Max Pooling	(64, 248, 98)	(64, 124, 49)	2
Convolution	(64, 124, 49)	(128, 122, 47)	(3, 3)
ReLU	(128, 122, 47)	(128, 122, 47)	-
Max Pooling	(128, 122, 47)	(128, 61, 23)	2
Convolution	(128, 61, 23)	(256, 59, 21)	(3, 3)
ReLU	(256, 59, 21)	(256, 59, 21)	-
Max Pooling	(256, 59, 21)	(256, 29, 10)	2

Table 9: Convolutional part of the gender classifier from full-body images.

Classifier

Layer	Size
Max Pooling Layer	-
Dense Layer	256
Dropout Layer 0.3	-
Dense Layer	64
Output	1

Table 10: Classifier structure for gender classification from full-body images.

As a detector, we can compare YOLOX-nano, that from Table 11 we can infer works at

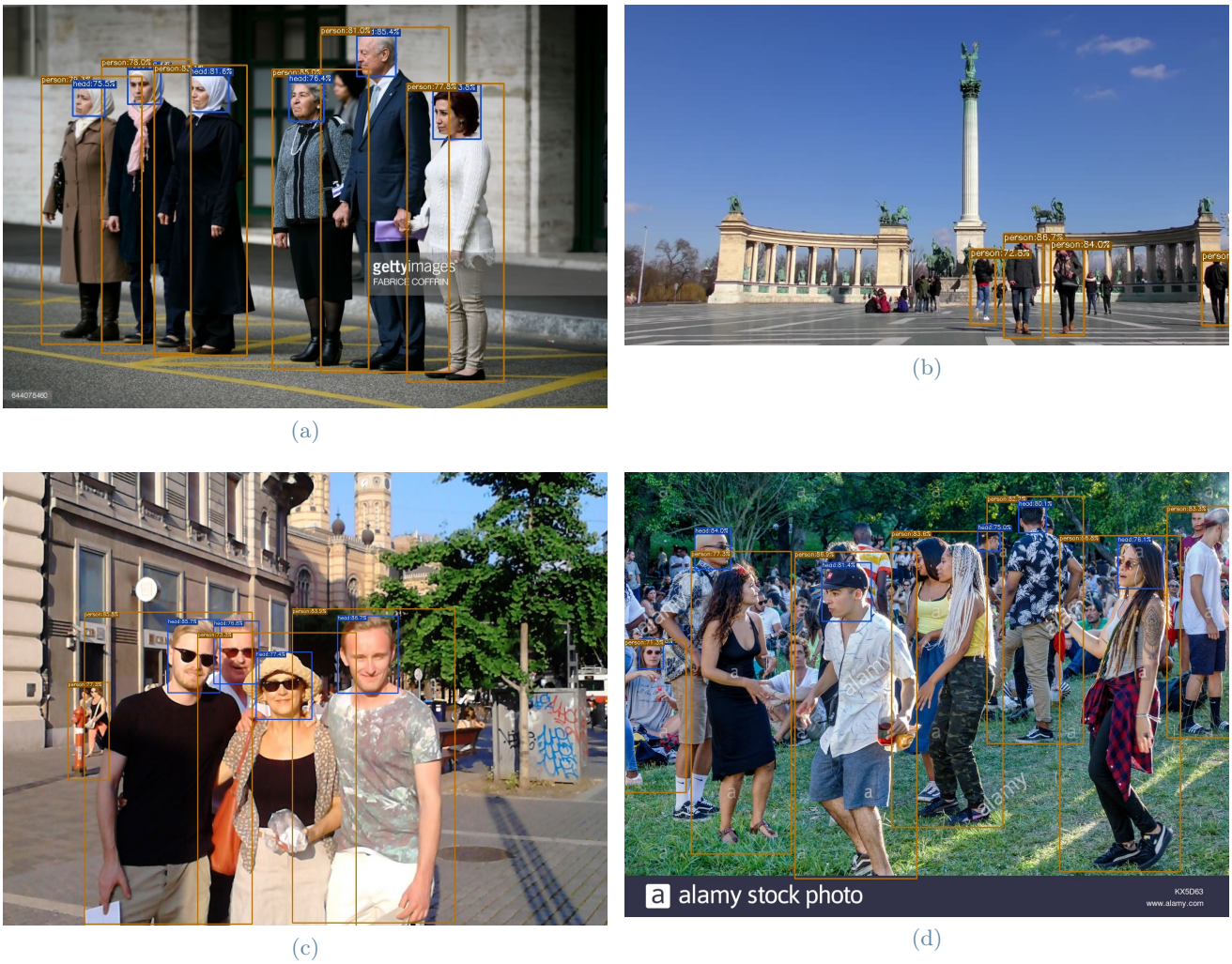
$$\text{FPS}_{YOLOX} = \frac{1}{0.1555} = 6.43 \text{ FPS} \tag{23}$$

while Tiny-YOLOv3 runs at 4.14 FPS. Greco et al. [16] proposed a real-time gender recognition model from face images that can operate at 5 FPS, while our Head Analysis module that performs both age and gender estimation does so at 12.3 FPS (using the same formula used for YOLOX).
An optimal solution is presented in [49], where a real-time single-shot multi-face gender detector based on a CNN can infer at 83 FPS. This suggests how a model where only one pass in the network is necessary can significantly increase the operating frequency.

Execution times (s)

Run	Detection	Association	Filter&Cut	Head Analysis	Body Analysis	Other	Total
1	0.1905	0.006	0.0188	0.0521	0.0529	0.0571	0.3774
2	0.2483	0.0042	0.0253	0.0728	0.0536	0.0242	0.4284
3	0.2074	0.0115	0.2308	0.0763	0.0521	0.1513	0.7294
4	0.3198	0.0054	0.0147	0.064	0.0632	0.0332	0.5003
5	0.1343	0.0053	0.0415	0.113	0.0541	0.0661	0.4143
6	0.0864	0.0063	0.0149	0.0764	0.0544	0.0351	0.2735
7	0.0977	0.0049	0.0155	0.1475	0.0618	0.0309	0.3583
8	0.0922	0.0048	0.0126	0.071	0.1196	0.2409	0.5411
9	0.0892	0.0054	0.0136	0.0642	0.0539	0.0646	0.2909
10	0.0896	0.0061	0.0211	0.0753	0.0613	0.1202	0.3736
AVG	**0.1555**	**0.006**	**0.0409**	**0.0813**	**0.0627**	**0.0824**	**0.4287**

Table 11: Execution times for different runs and average values. The detection is the most expensive operation, followed by analysis and filtering. The box association procedure is the least expensive. The resulting operating frequency obtained is 2.334 FPS.



(a) (b) (c) (d)

Figure 16: Examples of detections after fine-tuning YOLOX on the CrowdHuman dataset. Fewer subjects closer to the camera, (a) and (c), yield accurate detections. Crowded areas or distant subjects, (b) and (d), may lead to detection failure.

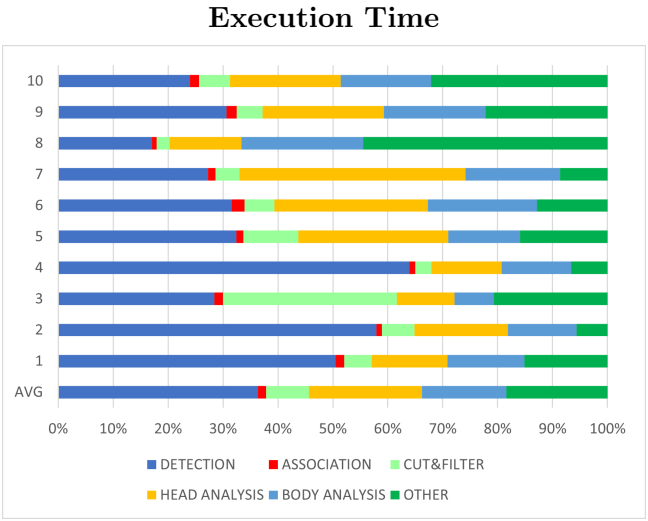


Figure 17: Execution times of the different tasks in ten runs, expressed in percentage.

In this section, the PAGE architecture has been described, starting from the general reasoning to the single components. The experiments carried out show how good performance is obtainable when performing gender regression from facial images, while it is worse when it is performed from full-body images. The age regression

models do not compare with the state-of-the-art and further discussion is in Section 4. The execution time is represented for the most part by the detection task, and the usage of different models can represent a reason for the low operating frequency. Even if running at low speed, it can be considered acceptable in a scenario that involves pedestrians where the movement speed is low. Comparing the single elements of the pipeline with other models, the inference time is comparable, but more time is necessary to handle the box association and preprocessing. Using a single-shot model represents a way to improve the computational speed.

4. Age Groups Classification

In this section, we discuss the efforts made to enhance age estimation through facial images, along with the various experiments conducted and the outcomes achieved. The ultimate goal was to create a model that could classify a facial image into one of the eight age groups, which are 0-9, 10-14, 15-24, 25-34, 35-54, 55-64 and 65+. The experiments are preceded by a discussion about the dataset considered and the data augmentation used during the training procedure. The tests investigate the effects of the input image size, the paradigm chosen to tackle the problem, and the size of the model. The section ends with a comparison of developed models with some state-of-the-art methods and the results obtained when using a new and challenging dataset.

4.1. Datasets

Before displaying the experiments carried out it is worth it to take into consideration the datasets used and analyse their composition to understand the motivations behind this section better.
The datasets considered are FGNET [13], UTKFace [66], AgeDB [35], APPA-real [2], KANFace [15], AAFD (All-Age-Faces Dataset) [7], and AFAD (Asian Face Age Dataset) [38]. In Table 12 the dataset sizes and age ranges are reported, while in Figure 18 and Figure 19 the plots representing the single ages and age groups distributions are displayed.

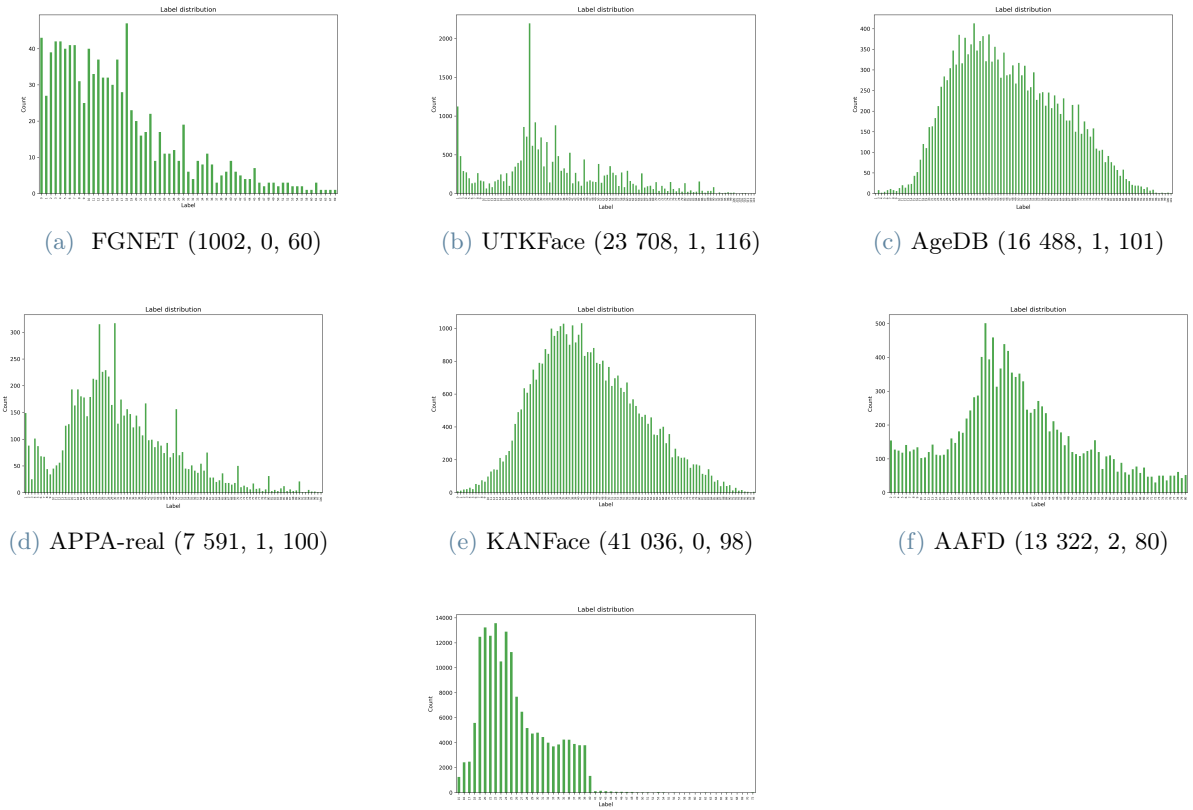
Dataset	Total Images	Age Range
FGNET	1 002	0 - 69
UTKFace	23 708	1 - 116
AgeDB	16 488	1 - 101
APPA-real	7 591	1 - 100
KANFace	41 036	0 - 98
AAFD	13 322	2 - 80
AFAD	165 501	15 - 72
TOTAL	**268 648**	**0 - 116**

Table 12: The datasets considered and their composition.

The difference in size and age ranges between the datasets is quite noticeable. For instance, AFAD contains 165k images whereas FGNET has only 1k. Moreover, UTKFace has the most age classes among all datasets, while AFAD has the least. The main issue is the class imbalance: all the datasets tend to have a higher concentration of samples in specific ranges, like AFAD where half of the images belong to the 15-24 range and no samples belong to the first two ranges or very few in the upper ranges. It is also possible to note a peak in the UTKFace age distribution in the 25-30 range and how the KANFace dataset is centred around 35 years of age. An exception is FGNET where the majority is represented by the 0-9 range but it is also the dataset with the smallest size. When the datasets are combined, these problems are worsened, as it is possible to notice in Figure 20. Most of the samples have labels in the range 18-35 and therefore 37.11% in the 15-24 class, 30.35% in the 25-34 range and 15.63% within the 35-44 range. Using this data to train a model would have the obvious consequence of overfitting the data inside those ranges and having very inaccurate predictions when the true age belongs to the other five range classes.
To try to mitigate this issue we decided to balance the data keeping only 5000 samples for each range. To do so we needed more samples in the 10-14 range since there are only 2506 elements in this class. To obtain more images we opted to create new ones starting from the ones already available and applying some methods of data augmentation. Specifically, we opted for applying a rotation range of 30°, a brightness range from 50% to 100%, and a horizontal flip. Examples of the data augmentation procedure on some images (not belonging to the 10-14 range) are shown in Figure 21. For each image another one was created, obtaining a total of 5012 samples for that range. Other ways of balancing the data have been to keep a fixed maximum number of samples per age value, for example, 1000, 1500 or 2000. In Figure 22 the age group distributions of these datasets are plotted.

4.2. Data Augmentation

To enhance the models' performance, as common, data augmentation has been used during training. The modifications applied include a brightness change from 50% to 150%, a channel shift with a range of 128, a maximum rotation of 30° and a zoom ranging from 0.75 to 1.0. One of the reasons for selecting this type of augmentation is that the datasets consist of images that have been captured under ideal conditions with good quality and



(a) FGNET (1002, 0, 60) (b) UTKFace (23 708, 1, 116) (c) AgeDB (16 488, 1, 101)

(d) APPA-real (7 591, 1, 100) (e) KANFace (41 036, 0, 98) (f) AAFD (13 322, 2, 80)

(g) AFAD (165 501, 15, 72)

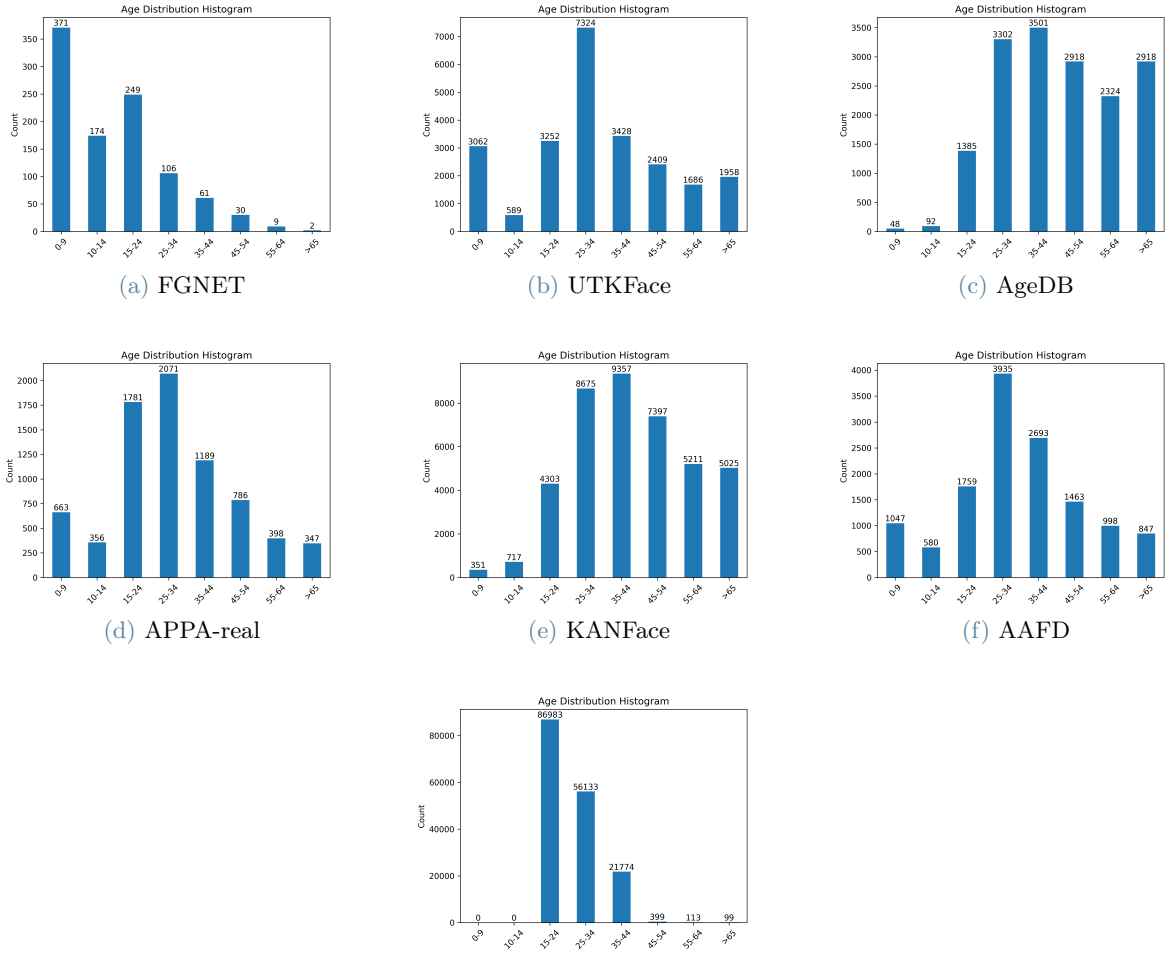
Figure 18: Datasets age distribution. The total number of samples and the minimum and maximum age values are reported under each dataset distribution with the format $(total\_samples, min\_age\_value, max\_age\_value)$.

optimal illumination. However, this work specifically focuses on a surveillance camera scenario, where the conditions can be challenging, with low resolution and variable illumination. Examples of the augmentation results are shown in Figure 23.

4.3. Input image size

The input image size plays an important role because it directly affects the feature size, i.e. the amount of information, that is extracted, influencing the quality of the predictions. State-of-the-art models usually consider input images to have at least a medium size, for example, 224 x 224 in [50, 64], 150 x 150 in [47] and 128 x 128 in [60]. Analysing the use case scenario of a HD surveillance camera we noticed how detected boxes in most cases have a small size, ranging from 50 x 50 to 100 x 100, therefore we selected 75 x 75 as a reasonable input size.

We investigated how much this reduction in the amount of features extracted can affect the prediction performance of a model. In Table 13 we report the MAEs on different datasets when used for training and testing. The model is composed of MobileNet V2 with an alpha value (that controls the network width) of 1.4 and three fully connected layers of 512 nodes. It is not surprising to find that as the input size increases, the mean absolute error (MAE) of the model decreases across all datasets. The minimum MAE decrease of 7% is observed on the AFAD dataset, while the maximum MAE improvement of 26% is seen on the dataset with a maximum of 5000 samples per age range. On average, the MAE decreases by 17%. In Figure 24 the confusion matrices when using 75 x 75 and 224 x 224 images are shown. With a higher resolution, the model improves on all the range classes besides the last one. The highest improvement is in the second class with a +16% in accuracy and the accuracy in the last range is decreased by 3%. On average the accuracy improves by 8.36%. It is worth focusing on the AFAD dataset, on which the model has the best MAE of 3.45 and 3.21. We can attribute this performance to the unbalance of the age distribution, already mentioned and plotted in Figure 18g. In Figure 25, the consequences of this phenomenon are shown. We can see in the confusion matrix that despite a low MAE, the actual accuracy is low when dealing with images whose true ages do not belong to the three most



(a) FGNET (b) UTKFace (c) AgeDB

(d) APPA-real (e) KANFace (f) AAFD

(g) AFAD

Figure 19: Datasets age groups distribution. Each bin represents how many samples in the dataset belong to that age range. It is possible to note how the higher concentration is in the 25-34 class and how some datasets like FGNET, KANFace and AFAD lack samples of the lower or higher classes.

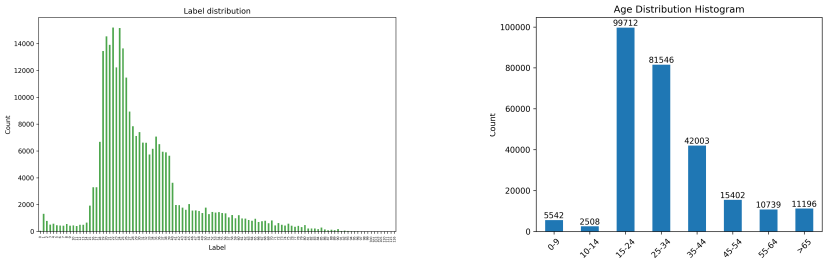
common classes. In the MAE per age diagram, it is clear how the MAE is low in the 20-30 range and gradually increases with higher age values.

4.4. Classification or Regression?

As explained in Section 3, age estimation can be approached in two ways: as a regression or as a classification problem. In the first experiment, we compared a classifier model, which directly classifies the input image into an age group, with a regression model. The regression model first tries to estimate the exact age and then translates it into the respective class. The conversion is carried out via a lambda layer that, given the estimated age, converts it into a vector as if it would have been the output of a standard classifier. The lambda layer applies a sequence of normal distribution, one for each age class so that the final vector has in each position the value of the corresponding distribution for that age. For each distribution, it is essential to define a mean and a standard deviation. In this work, those values have been chosen as:

$$\begin{aligned} mean &= \frac{x_1 + x_2}{2} \\ std &= \frac{x_2 - x_1}{2} \end{aligned} \tag{24}$$

where x_1 and x_2 are the lower and upper limits of each age group.
In the following Figure 26, the confusion matrices obtained from a typical regressor and a model explicitly trained to classify an image to a specific age range are compared. Both the models are trained using the dataset obtained with 5000 images per age range and the model structures are the same except for the last layer. It



(a) (268 648, 0, 116) (b)

Figure 20: Total age and group distribution. The age labels distribution is reported with the format $(total\_samples, min\_age\_value, max\_age\_value)$.

(a) from UTKFace (b) from FGNET

Figure 21: Examples of the procedure to obtain new images. To the original image, an augmentation with a rotation range of 30°, a brightness range from 50% to 100%, and a horizontal flip is applied. The resulting image is then saved.

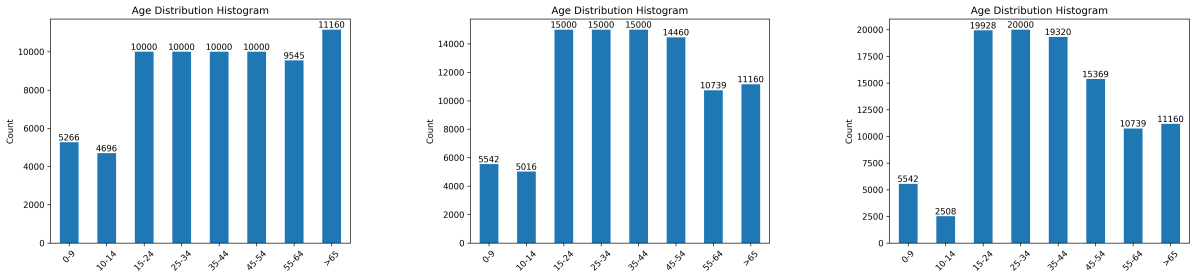
is clear how the best performance is given by the regressor, while the classifier is subject to completely wrong classifications, such as classifying images in the 25-34 range as a 0-9 class. Combining individuals of different ages into a single group can make it more challenging to identify shared characteristics, rather than simplifying the process. This approach can exacerbate the difficulty of finding common traits, which is already a major hurdle. Another major downside of a range classifier is that in case the desired age ranges are changed, the model has to be retrained, while with the regressor it is enough to change the lambda layer and the distributions at the end of the network. It was also verified whether tackling the problem as a classical classification problem, where each age is considered a class, could bring significant advantages. The resulting confusion matrices are available in Figure 27. The experiment illustrates how the classifier's performance can be improved for specific ranges and worsened for others, but given the very similar performance further tests would be necessary to affirm which paradigm is better with certainty. The next experiments use a regressor if not specified differently. The dataset used is the one with a maximum of 1500 images per age value and with a maximum age of 99. This cutoff has been selected because of too few elements above this age. The loss used to train the classifier for the age ranges is the Binary Cross Entropy, already introduced in Section 3 and for the regressor the Mean Squared Error (MSE):

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\text{true\_values}[i] - \text{predicted\_values}[i])^2 \tag{25}$$

where n is the number of samples, true_ages[i] predicted_ages[i] are the true and predicted values for the i-th sample. For the classifier with the single age classes, a custom loss has been developed to reproduce the MAE loss, called AbsoluteIndexDifference loss, where the difference between the class indices mimics the difference between age values since the class index 0 corresponds the the age value 0 and so on. The loss equation is available in Equation 26.

$$\text{AbsoluteIndexDifference} = \frac{\sum_{i=1}^{n} |\text{true\_indices}[i] - \text{pred\_indices}[i]| \cdot \text{sample\_weight}[i]}{\sum_{i=1}^{n} \text{sample\_weight}[i]} \tag{26}$$

where n is the number of samples,true_indices[i] and pred_indices[i] are the true and predicted indices for the i-th sample and sample_weight[i] is the weight assigned to the i-th sample (if provided). The weight chosen is 1 and equal for all the samples.



(a) max 1000 (b) max 1500 (c) max 2000

Figure 22: Different age groups distribution with different maximum number of samples for each age.

Figure 23: Original (left) and augmented (right) images used during training. The modifications involve adjusting brightness (50% to 150%), channel shifting (range of 128), rotation (maximum 30°), and zooming (0.75 to 1.0).

4.5. MobileNet

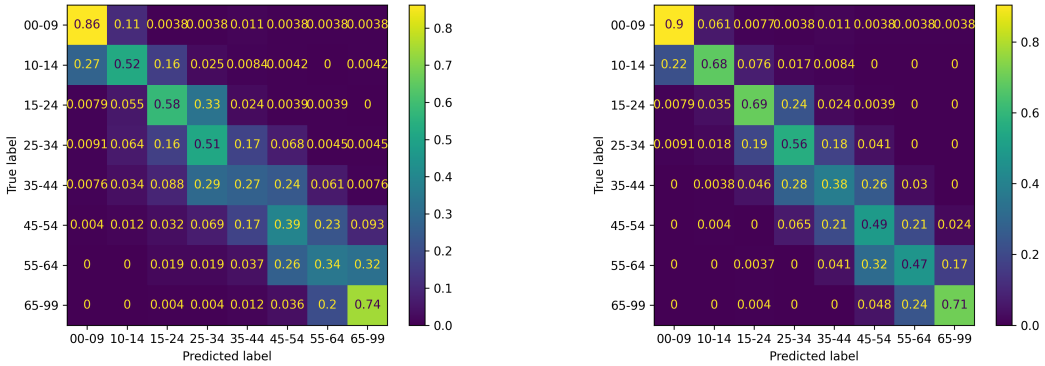
MobileNet [21] has been chosen as the baseline network because it is a very lightweight model and for its popularity in mobile applications where the computational power is limited.
In Figure 28, we can see the same model trained and tested on two datasets - one with 5000 images per range and the other with 1000 images per age value. Confusion matrices are shown in Figure 28a and Figure 28c, while MAE per age is depicted in Figure 28b and Figure 28d. The model is composed of the MobileNet V2 feature extractor followed by three fully connected layers, each with 512 nodes. Dropout layers with a 0.3 dropout rate are interchanged between them. Both models are configured as classifiers with 100 classes, from 0 to 99 years of age. Based on the analysis, we have observed that training with 1000 images per age value can lead to improved performance in specific ranges. For instance, in the age range of 15-24, a noticeable improvement of +10% has been registered, and in the age group of 65+, there is an improvement of +16%. However, we have noticed a decline in performance in the age ranges of 25-34 and 55-64. Focusing on the MAE per age diagrams, it is possible to note how the second model is better at predicting the highest age values, and this is probably due to the higher number of samples. The model with 5000 images per range has a MAE of 6.78, while the other has a MAE of 6.85, indicating similar overall performance. Another test verified how much the network width could improve the accuracy. The backbone selected is MobileNet V2 and the network width is regulated by the parameter *alpha*, which proportionally increases (when alpha > 1) or decreases (when alpha < 1) the number of filters in each layer. The default value for alpha is 1. Besides the MobileNet V2 backbone, the classifier is made of three fully connected layers of 1024 nodes interchanged with dropout layers with a dropout rate of 0.4. Between the feature extractor and the classifier, there is a GAP layer that aggregates each feature map to obtain a feature vector to be classified.

$$\mathrm{GAP}(F)_c = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} F_{i,j,c} \tag{27}$$

Where $\mathrm{GAP}(F)_c$ is the average value for channel c, $F_{i,j,c}$ is the value at position (i, j) in channel c, W is the width of the feature map, H is the height of the feature map, and C is the number of channels.
The dataset is composed of a maximum of 1000 samples per age value. The paradigm selected is a classifier with 100 age classes. The alpha values tested are 1, 1.3, 1.4 and 1.5 with resulting MAEs of 6.9, 6.62, 6.45 and 7.7 respectively. The confusion matrices and the MAE per age diagrams are reported in Figure 29. From the results obtained, we cannot sustain that increasing the model width guarantees a performance improvement.
A further test with MobileNet V2 investigated the impact of the maximum number of samples per age value using the datasets previously introduced. Besides the backbone with an alpha value of 1.4, the classifier was

Dataset	75 x 75	224 x 224
AAFD	6.97	5.25
AFAD	3.45	3.21
AgeDB	7.39	5.94
FGNET	6.64	6.07
KANFace	6.00	4.8
UTKFace	5.71	4.94
APPA-real	9.26	7.06
Max 5000 per range	6.87	5.09
Max 1500 per age	5.5	4.71

Table 13: Different datasets considered and the MAEs obtained with different input sizes. The AFAD dataset shows the minimum mean absolute error (MAE) decrease of 7%, while the dataset with a maximum of 5000 samples per age range exhibits the maximum MAE improvement of 26%. On average, there is a 17% decrease in MAE.



(a) 75 x 75

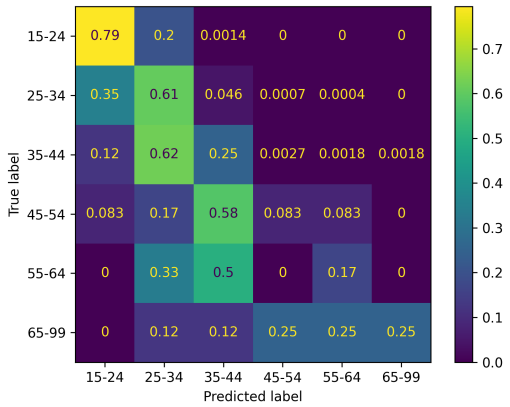
(b) 224 x 224

Figure 24: Confusion matrices comparing 75 x 75 and 224 x 224 input size on the dataset with a maximum of 5000 samples per range class. The most significant accuracy improvement occurs in the second class, with a notable increase of +16%, whereas the accuracy in the last range declines by 3%. On average, there is an overall accuracy improvement of 8.36%.

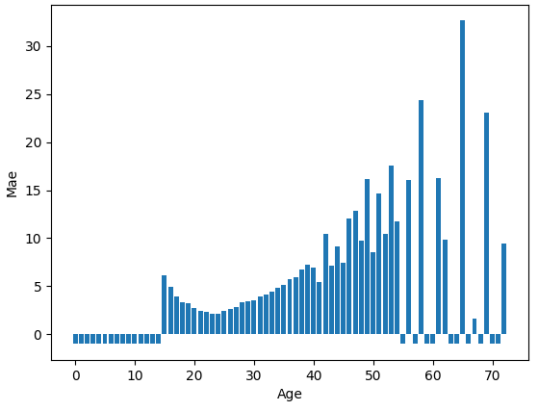
composed of three fully connected layers of 512 nodes each with a GAP layer in between. The chosen paradigm is the regressor. In Figure 30 the confusion matrices and the MAE per age diagrams are reported. The MAEs obtained are 5.86, 5.5 and 5.76 respectively. We can see how on average the best results are obtained when using a maximum of 1500 samples per age value, as it has the best accuracy on five out of eight range classes. The MAE per age is similar for each of the models, except for the high error in the first model on the 0 age, and the third model that has the highest MAE in the 15-20 range. The worst performance is reported at the higher end of the diagram for all the models.

4.6. Model Size

A different test performed released the constraint of using MobileNet as a baseline and it tried to investigate if employing a larger model would improve the prediction performance. The different backbones tested besides MobileNet V2 are VGG16 and VGG19 [51], ConvNeXt base and large [33], EfficientNet V2 Large [54], InceptionResnet V2 [53] and ResNet50 V2, ResNet101 V2 and ResNet152 V2 [20]. The different sizes of the networks are shown in Table 14 and the simple classifier is detailed in Table 15. The first layer of the classifier is a Global Average Pooling (GAP) layer. The dataset used is the one with 5000 images per age range. The detailed results are reported in Table 16. It is possible to note how larger models have on average a better accuracy, except for the 25-34 range where MobileNet has the best score. In particular, for some classes, the performance improvement can reach significant values for the baseline, like a +22% in the 10-14 range, +18% for the 15-24 class and +12% for 65+. Of course, the model size has to be taken into consideration in terms of



(a) Confusion Matrix



(b) MAE per Age. Negative values mean that those age values are not present in the test set

Figure 25: Confusion matrix and MAE per age diagram when the model is trained and tested on the AFAD dataset. It is clear how the imbalance on the 25-34 range impacts the accuracy when computed on the specific ranges. In the confusion matrix, the values tend to the second column and in the MAE per age diagram, the lowest columns correspond to the most common age values.

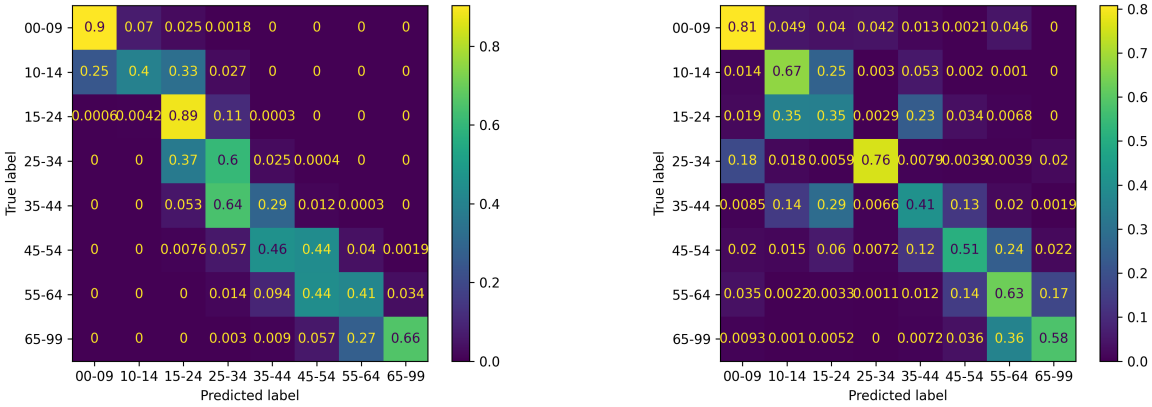
memory occupancy, training and inference times, but having larger models can result in better accuracy.

4.7. State of the art comparison

In this part of the section, a comparison between different state-of-the-art models and some of the developed models is reported. The developed models are two, one with MobileNet V2 with alpha 1.4 as the backbone and the other with VGG19 as the backbone. Both have three fully connected layers of 512 nodes as the classifier after the GAP layer. The results are available in Table 17. The proposed models report three results per database based on which dataset they have been trained. The first ones are the one with a max of 1000 per age, the second ones with a max of 1500 and the last with a max of 2000. The results are worse, but this is comprehensible given that our models are trained with all the datasets mixed, so they are not prone to overfit on the data of every different dataset, and they handle images with lower resolution (75 x 75), as already mentioned in this section.

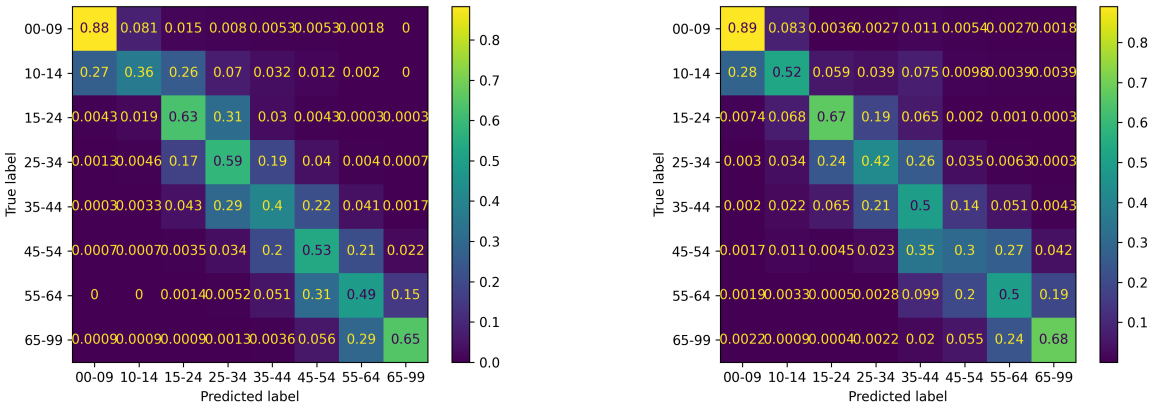
4.8. Challenging Data

The last series of tests carried out involved new data, acquired in a challenging scenario with a low-resolution camera in an office room. The samples are extracted from video sequences for a total of 267 images. This new dataset is therefore very limited in size and age values since the subjects are only six, each one with a different age. Another downside is that, being taken from a video sequence, most of the samples are taken from consecutive frames, therefore the images are very similar. For these reasons and to simulate a real-case scenario, the dataset has been used only as a test set. In Figure 31 some of the samples are shown and in Figure 32 the age and group distributions are plotted. The models tested are the same as the previous experiment. From Figure 33 it is possible to note how it is difficult to handle low-resolution images in a challenging scenario since every model has shown a significant decrease in performance. Specifically, we can note that the best model is the VGG 19 backbone trained with a maximum of 2000 samples per age value since it has obtained an accuracy of 31% on the 25-34 range and of 30% on the 35-44 class. The models that use MobileNet V2 as a backbone exhibit better performance with an increase in the number of samples, except for the 45-54 age group. While the first two classes start with an accuracy of 11% and 12%, they improve to 24% and 20%, respectively, and end with 32% and 24%. On the other hand, the accuracy of the last class starts at 17% and then decreases to 0.08%, finally ending up at 0%. With increasing samples, the models tend to improve average accuracy, as seen in the confusion matrices where the values shift towards the diagonal.



(a) Regressor (b) Range Classifier

Figure 26: Comparison between a regressor and a range classifier. The models have MobileNetV2 as the backbone and three fully connected layers of 512 nodes as the classifier. The regressor consistently demonstrates superior performance, whereas the classifier is prone to erroneous classifications, such as labelling images in the 25-34 age range as belonging to the 0-9 class.



(a) Regressor (b) Classifier

Figure 27: Comparison between a regressor and a classifier. The models have VGG16 as the backbone and four fully connected layers of 1024 nodes with dropout layers of 0.5 as the classifier. The experiment highlights the classifier's varied performance across different age ranges, suggesting potential for improvement in specific cases. However, due to the overall similar performance between the classifier and regressor, further tests are needed to conclusively determine the superior paradigm.

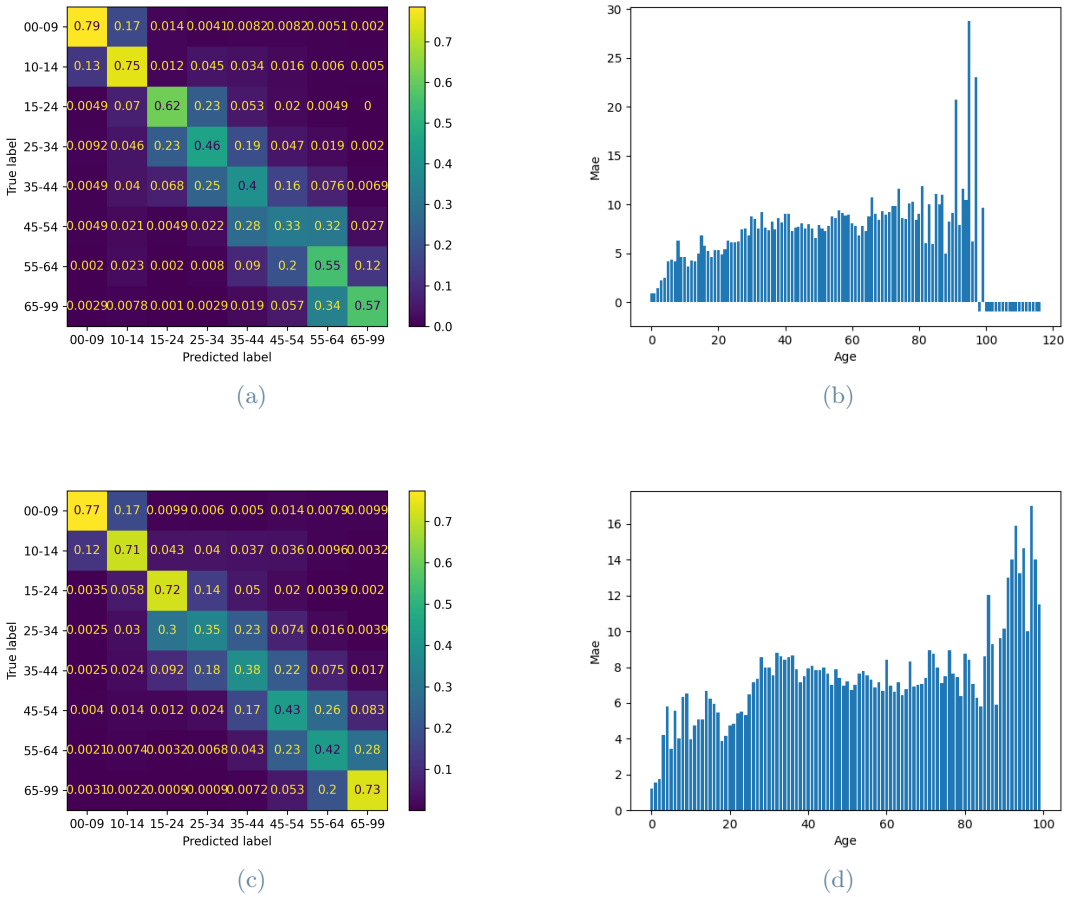
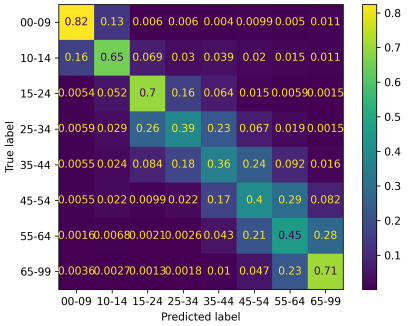
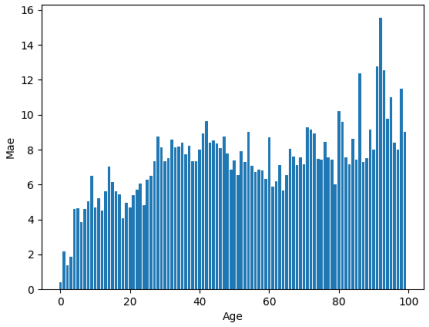


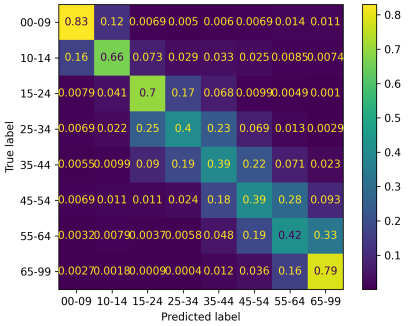
Figure 28: Comparison between using 5000 images per range or 1000 images per age value. The matrices report that training with 1000 images per age group can enhance performance in specific ranges, such as a notable +10% improvement in the 15-24 age range and a significant +16% improvement in the 65+ age group. However, performance declines in the 25-34 and 55-64 age ranges. Examining the mean absolute error (MAE) per age diagrams reveals that the second model excels in predicting higher age values, likely attributed to a higher number of samples. The model with 5000 images per range achieves a slightly lower MAE of 6.78, compared to the other model with a MAE of 6.85, indicating similar overall performance.



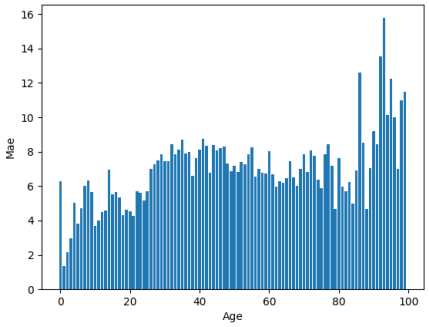
(a) alpha = 1



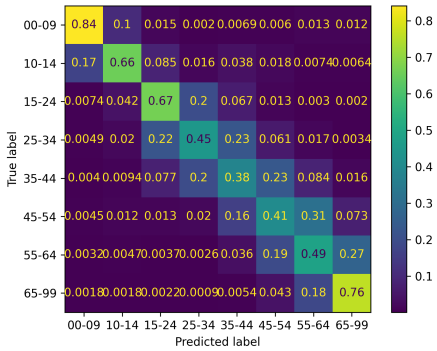
(b) alpha = 1



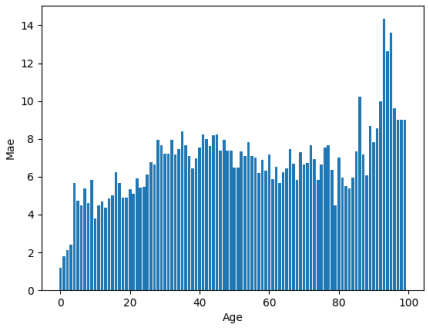
(c) alpha = 1.3



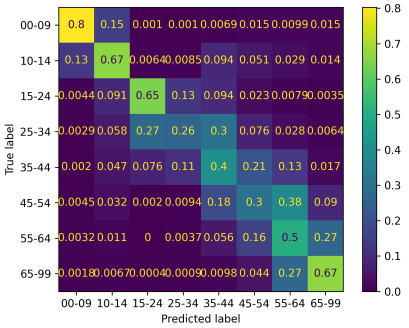
(d) alpha = 1.3



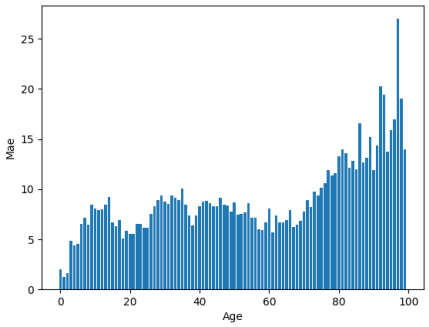
(e) alpha = 1.4



(f) alpha = 1.4

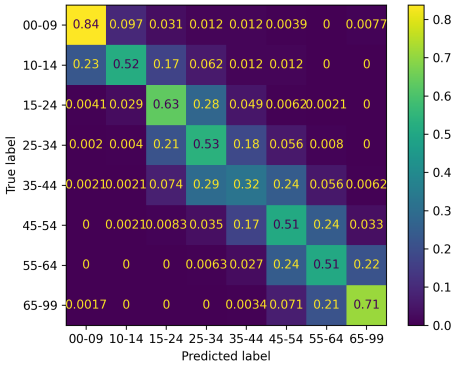


(g) alpha = 1.5

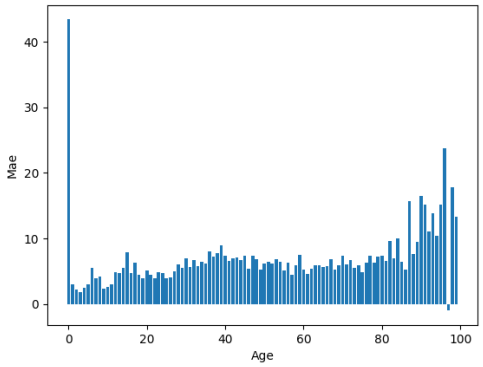


(h) alpha = 1.5

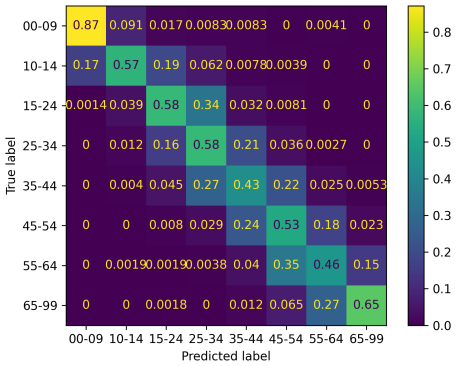
Figure 29: Comparison of models with MobileNet V2 backbone with different width values. Based on the obtained results, we cannot assert that augmenting the model width ensures an enhancement in performance.



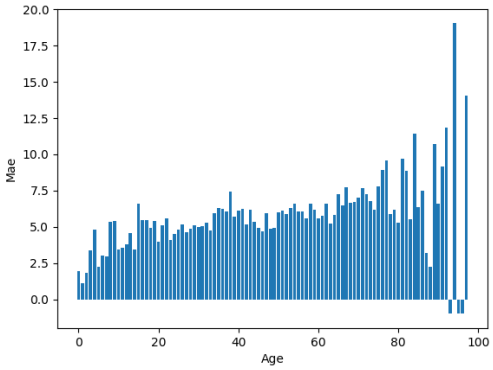
(a) max 1000 per age, MAE 5.86



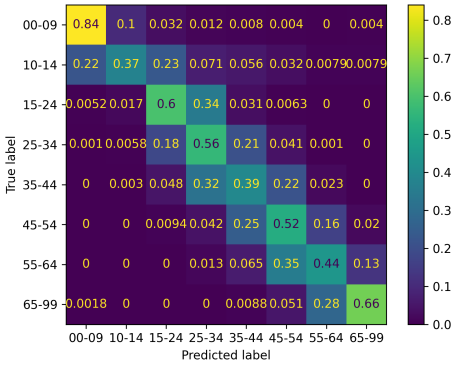
(b) max 1000 per age, MAE 5.86



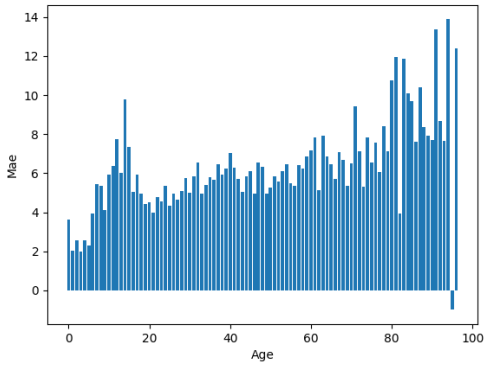
(c) max 1500 per age, MAE 5.5



(d) max 1500 per age, MAE 5.5



(e) max 2000 per age, MAE 5.76



(f) max 2000 per age, MAE 5.76

Figure 30: Comparison of models with MobileNet V2 backbone and different datasets with different maximum number of samples per age value. The mean absolute errors (MAEs) are recorded as 5.86, 5.5, and 5.76, respectively. Notably, the most favourable outcomes, with the highest accuracy in five out of eight range classes, are observed when utilizing a maximum of 1500 samples per age value. While the MAE per age generally exhibits similarities across models, exceptions include the first model's elevated error for age 0 and the third model's highest MAE in the 15-20 range. Consistently, suboptimal performance is reported at the higher end of the diagram for all models.

Feature Extractors

Model	Parameters
MobileNetV2 alpha 1	2,257,984
VGG16	14,714,688
VGG19	20,024,384
ConvNeXt base	87,566,464
ConvNeXt large	196,230,336
EfficientNetV2 large	117,746,848
InceptionResnetV2	54,336,736
ResNet50V2	23,564,800
ResNet101V2	42,626,560
ResNet152V2	58,331,648

Table 14: Different feature extractors and the number of parameters.

Classifier

Layer	Size
Global Average Pooling	0
Dense Layer	512
Dense Layer	512
Dense Layer	512
Output	1

Table 15: Classifier structure

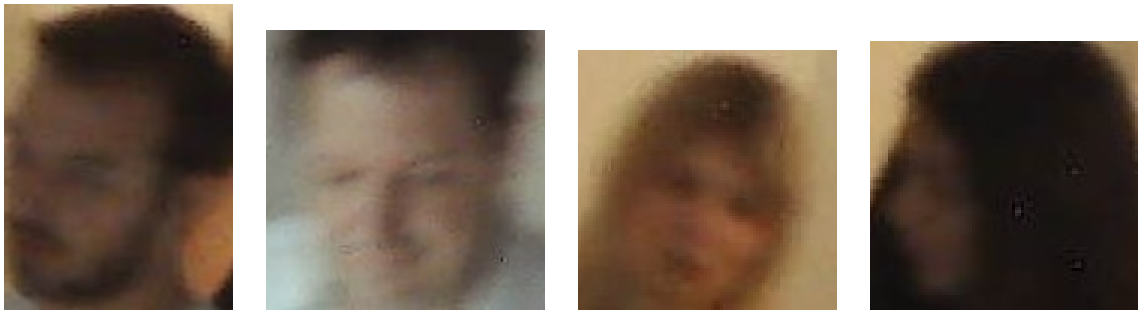
Network Size Results

Model	Params	0-9	10-14	15-24	25-34	35-44	45-54	55-64	65+	Average
MobileNetV2 alpha 1	2 257 984	0.8	0.48	0.5	**0.55**	0.33	0.44	0.41	0.59	0.5125
VGG16	14 714 688	0.85	0.67	0.55	0.52	0.36	0.49	0.51	0.64	0.5736
VGG19	20 024 384	0.83	0.61	0.6	0.49	0.37	0.49	0.49	0.68	0.57
ResNet50V2	23 564 800	0.83	0.53	0.6	0.5	0.27	0.43	0.44	0.59	0.5236
ResNet101V2	42 626 560	0.85	0.59	0.57	0.5	0.35	0.45	0.48	0.6	0.5488
InceptionResnetV2	54 336 736	0.86	0.57	0.58	0.47	0.34	0.47	0.47	0.55	0.5388
ResNet152V2	58 331 648	0.8	0.57	0.58	0.46	0.3	0.45	0.45	0.59	0.525
ConvNeXt base	87 566 464	0.86	**0.7**	**0.68**	0.51	0.27	**0.52**	**0.52**	0.6	0.5825
EfficientNetV2 large	117 746 848	0.84	0.67	0.58	0.53	**0.38**	0.48	**0.52**	**0.71**	**0.5888**
ConvNeXt large	196 230 336	**0.9**	0.61	0.63	0.52	0.29	0.5	0.51	0.67	0.5788

Table 16: Results of the network size test. The best results are in bold. Larger models generally exhibit improved accuracy on average, except in the 25-34 range where MobileNet achieves the highest score. Notably, significant performance enhancements compared to the baseline are observed, such as +22% in the 10-14 range, +18% for the 15-24 class, and +12% for the 65+ age group.

Model	FGNET	AFAD	UTKFace
Sharma et al. [47]	3.9	-	-
CDCNN [64]	-	3.11	-
MWR [50]	2.23	-	4.37
ADPF [59]	2.56	-	-
MobileNet V2 alpha 1.4	8.81 - 7.13 - 5.7	4.18 - 4.08 - 3.95	7.12 - 6.93 - 6.43
VGG19	8.03 - 7.33 - 6.86	4.06 - 3.97 - 3.83	6.69 - 6.38 - 6.01

Table 17: Comparison of SOTA models. The missing data means that the authors did not provide the results for that dataset. The obtained results are inferior, which is expected since the models are trained with a mixed dataset, preventing overfitting to specific datasets. Additionally, these models handle images with lower resolution (75 x 75), as mentioned earlier in this section.



(a) Age: 29 (95 x 127) (b) Age: 48 (139 x 139) (c) Age: 40 (89 x 89) (d) Age: 34 (86 x 86)

Figure 31: Examples of samples from the challenging dataset. The images are scaled for clarity and the original size and the label are reported under each one.

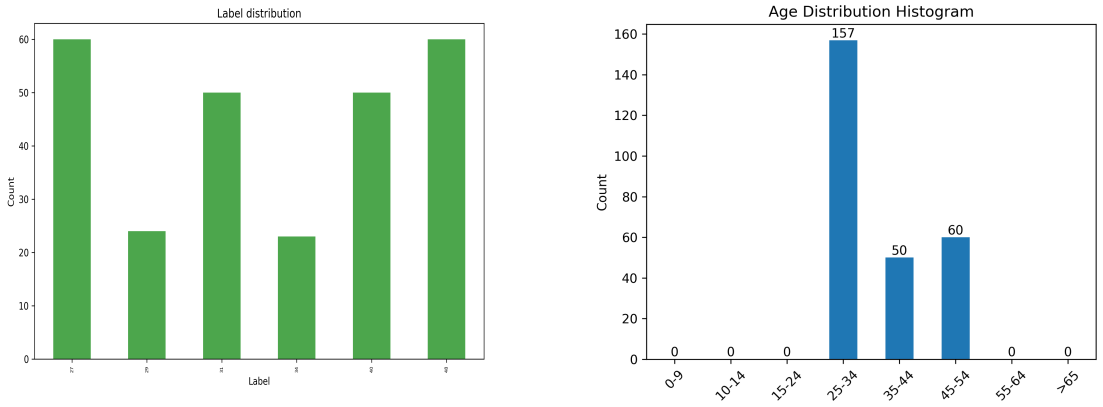
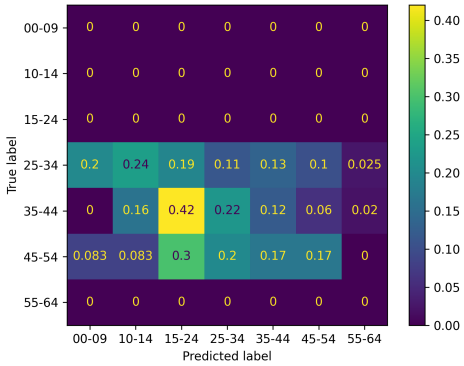
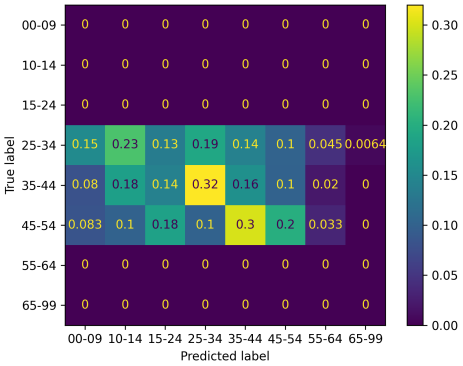


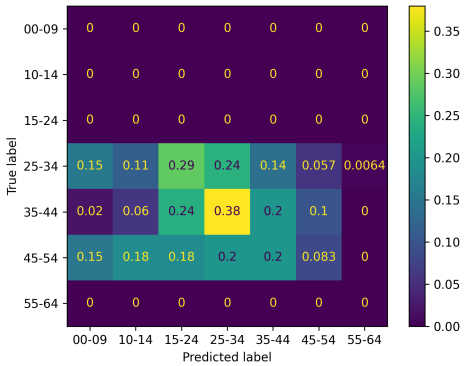
Figure 32: Challenging dataset distributions. There of 267 samples with age values ranging from 27 to 48.



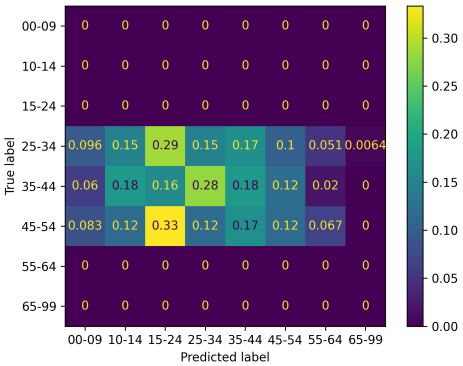
(a) MobileNet V2 backbone max 1000



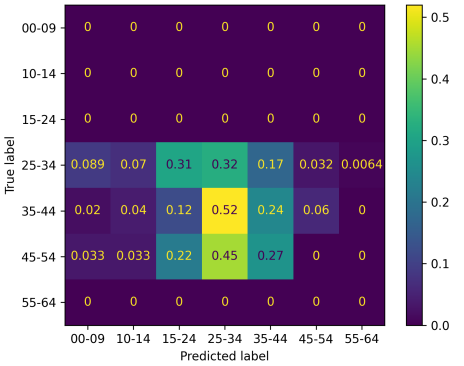
(b) VGG19 backbone max 1000



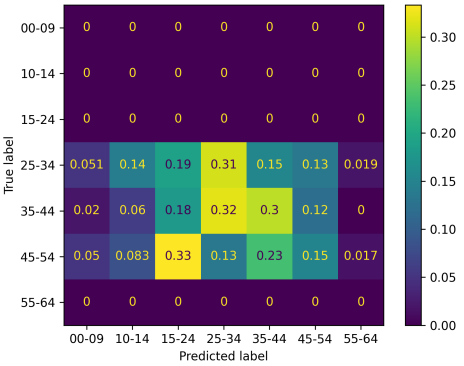
(c) MobileNet V2 backbone max 1500



(d) VGG19 backbone max 1500



(e) MobileNet V2 backbone max 2000



(f) VGG19 backbone max 2000

Figure 33: Test results on new and challenging data. The VGG 19 backbone trained with a maximum of 2000 samples per age value stands out as the best model, achieving 31% accuracy in the 25-34 range and 30% in the 35-44 class. Models utilizing MobileNet V2 as a backbone generally exhibit improved performance with an increase in the number of samples, except for the 45-54 age group. Increasing samples tends to enhance average accuracy, evident in the confusion matrices where values shift towards the diagonal.

5. Conclusions

In this work, we introduced PAGE (Pedestrian Age and Gender Estimation), a framework that combines pedestrian detection and age and gender estimation. The detections are obtained from the YOLOX detector and then processed by different models to obtain age and gender labels and to combine head and body boxes. The detector has been fine-tuned on the CrowdHuman dataset to add the head detection. Due to the limited size of the model, it has shown problems performing detection when there is a high density of subjects or when they are far away from the camera. The low resolution of the head and body images represents a significant obstacle to an accurate age and gender estimation. Analysing the operating time of the system, the heaviest operation has resulted in the detection, followed by the projection of the boxes and labels on the input image. Further work could be focused on creating a single end-to-end trainable model that performs both detection and age and gender estimation, besides trying different methods like hand-crafted features or using the attention mechanism to improve the prediction performance. Further research could be also about a more robust associating method, especially in the presence of overlaps.
The other goal of this project has been to try to develop an age estimation model that from a facial image could classify it into eight different age ranges. The most common datasets have been examined to get an understanding of their composition, and then different experiments have been performed. A strong augmentation is used to simulate the challenging scenario of a surveillance camera. They have shown how instead of using a model that directly classifies into the desired age ranges, it is more convenient to use a regressor and then remap the output to the corresponding class. It did not result in a better paradigm to use a regressor instead of a classifier with the same number of classes as age values, and the final answer is left for future studies. The input image size has emerged to be significant since, as expected, lower resolutions imply lower accuracy. Increasing the model size led to significant improvements in some classes but the consequences in terms of memory occupancy and inference time should be taken into consideration. Given the low resolution and strong augmentation used, the tested models' lower accuracy is justifiable compared to the state-of-the-art. When new and challenging data is tested, the models exhibit expectedly lower accuracy, but performance can be improved by using more samples. Future experiments could explore new CNN models, use attention mechanisms, or leverage different feature types.

References

[1] Farhat Abbas, Mussarat Yasmin, Muhammad Fayyaz, and Usman Asim. Vit-pgc: vision transformer for pedestrian gender classification on small-size dataset. *Pattern Analysis and Applications*, pages 1–15, 2023.

[2] Eirikur Agustsson, Radu Timofte, Sergio Escalera, Xavier Baro, Isabelle Guyon, and Rasmus Rothe. Apparent and real age estimation in still images with deep residual regressors on appa-real database. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pages 87–94. IEEE, 2017.

[3] Markus Braun, Sebastian Krebs, Fabian Flohr, and Dariu M Gavrila. Eurocity persons: A novel benchmark for person detection in traffic scenes. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1844–1861, 2019.

[4] Thomaz C. FEI Face Database. `https://fei.edu.br/~cet/facedatabase.html`, 2012.

[5] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018.

[6] Bor-Chun Chen, Chu-Song Chen, and Winston H. Hsu. Cross-age reference coding for age-invariant face recognition and retrieval. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[7] Jingchun Cheng, Yali Li, Jilong Wang, Le Yu, and Shengjin Wang. Exploiting effective facial patches for robust gender recognition. *Tsinghua Science and Technology*, 24(3):333–345, 2019.

[8] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005.

[9] Patrick Dendorfer, Hamid Rezatofighi, Anton Milan, Javen Shi, Daniel Cremers, Ian Reid, Stefan Roth, Konrad Schindler, and Laura Leal-Taixé. Mot20: A benchmark for multi object tracking in crowded scenes. *arXiv preprint arXiv:2003.09003*, 2020.

[10] Yubin Deng, Ping Luo, Chen Change Loy, and Xiaoou Tang. Pedestrian attribute recognition at far distance. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 789–792, 2014.

[11] Mark Everingham and John Winn. The pascal visual object classes challenge 2012 (voc2012) development kit. *Pattern Analysis, Statistical Modelling and Computational Learning, Tech. Rep*, 8(5):2–5, 2011.

[12] Muhammad Fayyaz, Mussarat Yasmin, Muhammad Sharif, and Mudassar Raza. J-ldfr: joint low-level and deep neural network feature representations for pedestrian gender classification. *Neural Computing and Applications*, 33:361–391, 2021.

[13] Yanwei Fu, Timothy M Hospedales, Tao Xiang, Jiechao Xiong, Shaogang Gong, Yizhou Wang, and Yuan Yao. Robust subjective visual property prediction from crowdsourced pairwise labels. *IEEE transactions on pattern analysis and machine intelligence*, 38(3):563–577, 2015.

[14] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021.

[15] Markos Georgopoulos, Yannis Panagakis, and Maja Pantic. Investigating bias in deep face analysis: The kanface dataset and empirical study. *Image and vision computing*, 102:103954, 2020.

[16] Antonio Greco, Alessia Saggese, and Mario Vento. Digital signage by real-time gender recognition from face images. In *2020 IEEE International Workshop on Metrology for Industry 4.0 and IoT*, pages 309–313, 2020.

[17] Jing Han, Xiaoying Wang, Xichang Wang, and Xueqiang Lv. Cfnet: Head detection network based on multi-layer feature fusion and attention mechanism. *IET Image Processing*, 17(7):2032–2042, 2023.

[18] Irtiza Hasan, Shengcai Liao, Jinpeng Li, Saad Ullah Akram, and Ling Shao. Generalizable Pedestrian Detection: The Elephant in the Room. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 11323–11332, 2021.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 630–645. Springer, 2016.

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

[22] Wei Yen Hsu and Wen Yen Lin. Ratio-and-Scale-Aware YOLO for Pedestrian Detection. *IEEE Transactions on Image Processing*, 30:934–947, 2021.

[23] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Workshop on faces in'Real-Life'Images: detection, alignment, and recognition*, 2008.

[24] Hyunduk Kim, Myoung Kyu Sohn, and Sang Heon Lee. Development of a Real-Time Automatic Passenger Counting System using Head Detection Based on Deep Learning. *Journal of Information Processing Systems*, 18(3):428–442, 2022.

[25] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[26] Seung Hoon Lee, Seunghyun Lee, and Byung Cheol Song. Vision transformer for small-size datasets. *arXiv preprint arXiv:2112.13492*, 2021.

[27] Shengcai Liao, Yang Hu, Xiangyu Zhu, and Stan Z Li. Person re-identification by local maximal occurrence representation and metric learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2197–2206, 2015.

[28] Shengcai Liao, Guoying Zhao, Vili Kellokumpu, Matti Pietikäinen, and Stan Z Li. Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 1301–1306. IEEE, 2010.

[29] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[30] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8759–8768, 2018.

[31] Xihui Liu, Haiyu Zhao, Maoqing Tian, Lu Sheng, Jing Shao, Shuai Yi, Junjie Yan, and Xiaogang Wang. Hydraplus-net: Attentive deep features for pedestrian analysis. In *Proceedings of the IEEE international conference on computer vision*, pages 350–359, 2017.

[32] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[33] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022.

[34] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016.

[35] Stylianos Moschoglou, Athanasios Papaioannou, Christos Sagonas, Jiankang Deng, Irene Kotsia, and Stefanos Zafeiriou. Agedb: the first manually collected, in-the-wild age database. In *proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 51–59, 2017.

[36] Alexander Neubeck and Luc Van Gool. Efficient non-maximum suppression. In *18th international conference on pattern recognition (ICPR'06)*, volume 3, pages 850–855. IEEE, 2006.

[37] Zhaoyang Niu, Guoqiang Zhong, and Hui Yu. A review on the attention mechanism of deep learning. *Neurocomputing*, 452:48–62, 2021.

[38] Zhenxing Niu, Mo Zhou, Le Wang, Xinbo Gao, and Gang Hua. Ordinal regression with multiple output cnn for age estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4920–4928, 2016.

[39] Constantine P. Papageorgiou, Michael Oren, and Tomaso Poggio. General framework for object detection. *Proceedings of the IEEE International Conference on Computer Vision*, pages 555–562, 1998.

[40] Dezhi Peng, Zikai Sun, Zirong Chen, Zirui Cai, Lele Xie, and Lianwen Jin. Detecting heads using feature refine net and cascaded multi-scale architecture. *arXiv preprint arXiv:1803.09256*, 2018.

[41] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-December:779–788, 2016.

[42] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[43] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[44] Karl Ricanek and Tamirat Tesafaye. Morph: A longitudinal image database of normal adult age-progression. In *7th international conference on automatic face and gesture recognition (FGR06)*, pages 341–345. IEEE, 2006.

[45] Rasmus Rothe, Radu Timofte, and Luc Van Gool. Dex: Deep expectation of apparent age from a single image. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*, December 2015.

[46] Shuai Shao, Zijian Zhao, Boxun Li, Tete Xiao, Gang Yu, Xiangyu Zhang, and Jian Sun. Crowdhuman: A benchmark for detecting human in a crowd. *arXiv preprint arXiv:1805.00123*, 2018.

[47] Neha Sharma, Reecha Sharma, and Neeru Jindal. Face-Based Age and Gender Estimation Using Improved Convolutional Neural Network Approach. *Wireless Personal Communications*, 124(4):3035–3054, 2022.

[48] Mohammadreza Sheikh Fathollahi and Rezvan Heidari. Gender classification from face images using central difference convolutional networks. *International Journal of Multimedia Information Retrieval*, 11(4):695–703, 2022.

[49] Tak Wai Shen, Dongpeng Wang, Kayton Wai Keung Cheung, Man Chi Chan, King Hung Chiu, and Yiu Kei Li. A real-time single-shot multi-face detection, landmark localization, and gender classification. In *Proceedings of the 2021 3rd International Conference on Image Processing and Machine Vision*, pages 1–4, 2021.

[50] Nyeong Ho Shin, Seon Ho Lee, and Chang Su Kim. Moving Window Regression: A Novel Approach to Ordinal Regression. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2022-June:18739–18748, 2022.

[51] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[52] Zhihong Sun, Jun Chen, Mithun Mukherjee, Haihui Wang, and Dang Zhang. An Improved Online Multiple Pedestrian Tracking Based on Head and Body Detection. *Proceedings - 2021 17th International Conference on Mobility, Sensing and Networking, MSN 2021*, pages 74–80, 2021.

[53] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 4278–4284. AAAI Press, 2017.

[54] Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *International conference on machine learning*, pages 10096–10106. PMLR, 2021.

[55] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[56] Paul Viola, Michael Jones, et al. Robust real-time object detection. *International journal of computer vision*, 4(34-47):4, 2001.

[57] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7464–7475, 2023.

[58] Chien-Yao Wang, Hong-Yuan Mark Liao, Yueh-Hua Wu, Ping-Yang Chen, Jun-Wei Hsieh, and I-Hau Yeh. Cspnet: A new backbone that can enhance learning capability of cnn. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 390–391, 2020.

[59] Haoyi Wang, Victor Sanchez, and Chang Tsun Li. Improving Face-Based Age Estimation with Attention-Based Dynamic Patch Fusion. *IEEE Transactions on Image Processing*, 31:1084–1096, 2022.

[60] Xiao Wang, Shaofei Zheng, Rui Yang, Aihua Zheng, Zhe Chen, Jin Tang, and Bin Luo. Pedestrian attribute recognition: A survey. *Pattern Recognition*, 121:108220, 2022.

[61] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[62] Lei Yang, Guowu Yuan, Hao Zhou, Hongyu Liu, Jian Chen, and Hao Wu. Rs-yolox: A high-precision detector for object detection in satellite remote sensing images. *Applied Sciences*, 12(17):8707, 2022.

[63] Zitong Yu, Chenxu Zhao, Zezheng Wang, Yunxiao Qin, Zhuo Su, Xiaobai Li, Feng Zhou, and Guoying Zhao. Searching central difference convolutional networks for face anti-spoofing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5295–5305, 2020.

[64] Beichen Zhang and Yue Bao. Cross-Dataset Learning for Age Estimation. *IEEE Access*, 10:24048–24055, 2022.

[65] Shanshan Zhang, Rodrigo Benenson, and Bernt Schiele. Citypersons: A diverse dataset for pedestrian detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3221, 2017.

[66] Zhifei Zhang, Yang Song, and Hairong Qi. Age progression/regression by conditional adversarial autoencoder. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5810–5818, 2017.

Abstract in lingua italiana

L'analisi dei pedoni è utilizzata in ambiti molto diversi, da applicazioni di sicurezza all'analisi del traffico pedonale o per altri scopi commerciali. Per effettuare questa operazione è necessario in primo luogo rilevare i pedoni per poi estrarne delle informazioni utili. In questo progetto è presentato PAGE (Pedestrian Age and Gender Estimation), un semplice metodo che combina la rilevazione dei pedoni e la stima di genere ed età. PAGE sfrutta le rilevazioni di YOLOX e diversi modelli per classificare immagini di teste e corpi a figura intera a bassa risoluzione. Gli esperimenti effettuati hanno mostrato come questo metodo può essere considerato un promettente primo passo rispetto a futuri sviluppi sull'analisi dei pedoni. L'altro apporto di questo lavoro è rappresentato da degli esprimenti mirati a migliorare la precisione di un modello per la stima dell'età che deve classificare l'immagine secondo otto scaglioni di età. Come modello di base è stata scelta MobileNet, visto il suo buon compromesso fra dimensioni e precisione. Usare un regressore o un classificatore in cui ogni classe è un'età si è dimostrato meglio di utilizzare un classificatore dove ogni classe è uno scaglione di età, sia in termini di precisione che di flessibilità. Altri test hanno verificato come la risoluzione delle immagini e le dimensioni del modelli giochino un ruolo essenziale per ottenere delle stime più accurate dell'età di un individuo.

Parole chiave: rilevamento pedoni, stima dell'età, stima del genere

Ringraziamenti

Vorrei innanzitutto esprimere la mia profonda gratitudine al professor Matteo Matteucci che mi ha dato l'opportunità di sviluppare questo progetto di tesi e a Simone Mentasti che, come correlatore, mi ha seguito e consigliato durante tutto il percorso. Un grazie va a tutta la mia famiglia, in particolare ai miei genitori, che mi hanno permesso di compiere questo percorso di studi che mi ha accresciuto dal punto di vista accademico ma soprattutto personale. Grazie agli amici di sempre con cui posso sempre passare una serata spensierata e che si sforzano di ridere alle mie battute. Grazie agli amici che ho conosciuto durante questi cinque anni, per i momenti felici ma anche quelli più difficili che abbiamo condiviso dentro e fuori il Poli. Un ringraziamento sentito alla prof.ssa Muggiasca, che nel momento in cui avevo più dubbi sul mio futuro mi ha aiutato a capire che questa fosse la mia strada, ma anche a tutti gli insegnati e professori che ho incontrato in questi anni. Grazie a Emma, la persona che mi è stata più vicina e che mi ha sempre sostenuto, anche quando il mio percorso ci ha separato.