FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Age Estimation using Deep Learning on 3D Facial Features

Pedro Vieira de Castro

U. PORTO

FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Dr. Yifan Zhao

Co-Supervisor: Dr. Luís Teixeira

September 22, 2018

Age Estimation using Deep Learning on 3D Facial Features

Pedro Vieira de Castro

Mestrado Integrado em Engenharia Informática e Computação

September 22, 2018

Abstract

Intelligent Systems are designed to substitute the human component therefore they have a need to emulate a human's ability to quickly estimate biological traits of others, which is an integral part of social interactions. Age is one of the key characteristics used by marketing, entertainment and security tools.

Existing age estimation systems can be easily fooled due to their reliance on human appearance based features, which can be easily manipulated. Over the years, while the complexity of models increased, the data fed to our systems was kept the same: a single 2D RGB image. This thesis addresses the current lack of studies made on the uses of 3D facial information on the context of age estimation.

This thesis encompasses a comprehensive study of how different 3D facial features can be used to improve current state of the art age estimation approaches using Deep Learning. Along with extensions to a baseline Convolutional Neural Network (CNN) model with a 2D image input, it is introduced a novel Multi-View CNN model which combines face descriptors from multiple perspectives within the model's architecture.

Due to lack of 3D facial datasets aimed at age estimation, 2D age estimation datasets were synthetically augmented with landmark localization, 3DMM parametrization and 3D facial point cloud reconstruction. The last one was subsequently used to create a new synthetic dataset composed of renderings of each point cloud from different camera positions. A fully customizable data processing tool was introduced which supports image pre-processing, dataset splitting, image augmentation and synthetic feature extraction.

Quantitative results show improvement of the 3D methods over traditional 2D although somewhat constrained by data quality.

Resumo

Sistemas inteligentes são desenvolvidos a fim de substituírem componentes humanos existentes. A fim de atingirem este objectivo, estes sistemas requerem a habilidade de estimar características biológicas dos seus utilizadores, de forma a emular aptidões humanas presentes em qualquer interacção social. A capacidade de estimar a idade de um utilizador é um recurso valioso para qualquer ferramenta de marketing, entretenimento ou segurança digital.

Sistemas actuais de estimação de idade são facilmente iludidos devido à sua dependência em características faciais facilmente manipuláveis com maquilhagem, operações plásticas, bronzeado, etc. Estimação de idade a partir de imagens 2D é um tópico que tem sido estudado a fundo nos últimos anos no entanto o uso de características 3D tem sido negligenciado devido tanto à falta de dados como à fraca capacidade de os gerar. Esta tese foca-se no estudo do impacto do uso de características 3D de forma a melhorar métodos de estimação de idade actuais. Nesta tese, é apresentado um estudo comparando os diferentes impactos que diversas características 3D faciais têm na tarefa de estimação de idade.

São apresentadas melhorias a actuais modelos Deep Learning, como extensões a modelos convolucionais. Para além de extensões a modelos existentes, é introduzido um modelo *Multi-View* que combina vários descritores faciais, baseados em diferentes projecções da mesma face, num só descritor.

Visto que os datasets usualmente estudados por esta área de investigação são puramente 2D, estes são estendidos sinteticamente de forma a obter características 3D tal como pontos de referência faciais, 3DMM, *Point Clouds* e rasterizacão destes em imagens 2D, usando métodos estado da arte disponíveis. Para além do estudo anterior, também é introduzida uma ferramenta capaz de gerar diversa informação 3D partindo de apenas uma imagem 2D, incluindo pré-processamento, divisão de datasets, manipulação de imagens e extracção de dados 3D.

Resultados quantitativos mostram uma melhoria quando adicionados elementos 3D à CNN original. Porém este incremento na eficácia do modelo aparenta ser proporcional à qualidade da imagem original, qualidade que se propaga pelos respectivos dados sintéticos e rasterizações.

Acknowledgements

First of all, I would like to thank my father who has always loved and supported me no matter what. Thank you!

I would like to thank Jaguar Land Rover, specially the Research team, for their support and advice as well as for allowing me to use their machines during downtime.

I would also like to thank Dr. Karl Ricanek Jr., director of the Face Ageing Group, from the University of North Carolina Wilmington, who provided one of the datasets used in this work free of cost.

Lastly, I would like to thank my supervisor Dr. Yifan Zhao, from the University of Cranfield, for the continuous support throughout the making and writing of this thesis.

Pedro Castro

Contents

List of Figures

List of Tables

LIST OF TABLES

Abbreviations

2D	Two Dimensional Type
3D	Three Dimensional
CNN	Convolutional Neural Network
SGD	Stochastic Gradient Descent
PCA	Principal Component Analysis
3DMM	3D Morphable Model
MAE	Mean Average Error
ReLU	Rectified Linear Unit
ADAM	Adaptive Moment Estimation
SVR	Support Vector Regression
AGES	AGeing pattErn Subspace
BIF	Bio-Inspired Features
LBP	Local Binary Patterns
LSTM	Long-Short Term Memory
LAP	Looking at People
ICCV	International Conference on Computer Vision
ASM	Active Shape Model
AOM	Active Orientation Model
LOO	Leave One Out

Chapter 1

Introduction

1.1 Motivation

The task of age estimation is a key component of an intelligent expert system, whose objective is to substitute the human component in many information systems from a variety of fields. Security focused tools can age restrict sensitive content from underaged users, whether this content is purchasable or viewable. Marketing tools steer personalized advertisement based on age information reducing the big overhead costs of advertising a product to the wrong target audience. Age information can be applied in many other sectors such as law enforcement, human-control interaction and social media.

Every human is equipped with the ability of estimating the age of someone with a simple face glance. It is similar to other biological trait detection tasks which humans are well versed on such as gender and ethnicity classification, health status diagnostic and facial expression interpretation, which are possible due to the existence of non-verbal characteristics. These innate human capabilities are essential every day tools and are an integral part of social interactions.

However, even humans can be easily fooled by other non-natural factors such as makeup, skin tan, both hair facial style, plastic surgery, among many others. Fashion changes dictate what kind of hairstyle young and old wear, hair dye products are cheap and easily obtainable and so is access to skin tanning products or even the beach. All these factors can influence someone's perceived age and contribute to a more complex age estimation task.

Since current age estimation systems are appearance based, it is hard to tell if intelligent systems also fall into the same pitfalls. The same problem occurs in face identification systems that must be robust to slight facial changes that may occur [SKP15]. The current state of the art on face identification has become sufficient for it to enter mainstream products. For example, we are currently seeing the market of mobile phones switching from fingerprint identification to face identification as the preferred method for unlocking a secure device, catching the public's attention with the introduction of Apple's IPhone X in 2017.

Although extensive research on facial age estimation, the use of the 3D shape of the face has been neglected from most studies. Scientists are aware that human facial skeleton changes with age in particular the midface area, nose, eye sockets and jaw [MW12]. For this reason, we think it is important to take into consideration the facial structure, which contains 3D information, of a person when estimating one's age, in addition to that person's facial appearance which is commonly represented as a 2D image.

A lot of research age estimation through facial imaging has been done over the last years. Contests have been organized [EFP+15] in order to encourage researchers to delve into this field. Results took a huge leap in effectiveness with the introduction of deep learning, specially through the use of Convolutional Neural Networks (CNN).

1.2 Aim and Objectives

In this thesis, we will study how we can use 3D facial information to enhance age estimation from a single image. We aim to prove that improvements can be made on this research field by enhancing the existing state of the art approaches on 2D facial age estimation with the addition of 3D features representations.

The objectives of this thesis are:

- Study the current state of the art Deep Learning architectures, age estimation systems and techniques and 3D face modelling methods.
- Create a tool to collect, clean and pre-process the most commonly used age estimation publicly available datasets in a seamless way and fully personalized way.
- Create a CNN model which takes as input a single 2D image to serve as a baseline comparison for our future studies.
- Extend the collected 2D datasets with synthetically extracted 3D facial structure information, due to the lack of 3D facial datasets aimed at the age estimation task and incorporate these augmentations as part of the data processing pipeline previously mentioned.
- Extend the baseline model to take as additional inputs the extracted 3D facial information therefore creating multiple models with different architectures and input structures.
- Do a comparison study between the different models and different available 3D information to demonstrate the advantages and disadvantages of each when compared to the baseline.
- Perform a cross-dataset evaluation of the models to determine how well these methods are able to generalize to other data distributions.
- Understand the difficulties of this task by extending the error analysis to the prediction error distributions and accuracy metrics on age ranges.

Chapter 2

Convolutional Neural Networks

2.1 Introduction

We can broadly categorize machine learning tasks into three types: Supervised Learning, Unsupervised Learning and Reinforcement Learning.

The first one, the most common form and the one that will be used throughout this thesis, is a task that consists on learning a function that maps a set of independent variables, i.e. the input, to a desired output, or a dependent variable. For this type of learning it is vital that we possess enough examples of input/output pairs on which the algorithm will train on. This paradigm's key factor for a successful supervised learning algorithm is the algorithm's ability to generalize to unseen data.

For many years, creating a machine-learning system required careful engineering and an extensive expert knowledge to design an efficient feature extractor to transform raw data (such as characters of a text or pixels of an image) into a representation or abstraction, of which the underlying system could learn patterns from.

Deep learning introduced representation learning with multiple levels of abstraction, obtained by composing simple but non-linear modules capable of transforming the data representation at one level into a representation level with a higher level of abstraction. When an image, for example, is fed into a deep learning algorithm, the learned features on the first representation layers will most likely represent the presence or absence of low level features such as edges, at certain location and direction. Subsequent layers will detect combinations of this edges that will in eventually create representations of textures and objects. By layering together enough transformations, it is possible to learn very complex functions. And by learning feature representation, features are no longer designed by humans and are learned directly from data [LBH15].

Convolutional Neural Networks were first introduced by [FM82] and later refined by [LHBB99], who first applied Back Propagation as the training algorithm. These models were inspired by neural connections, hence the name, and are one of the most successful applications of the knowledge gathered by studying the brain. These models take advantage of structured data with spatially relevant information, whether it be audio (1D), images (2D) or volumetric images(3D) for example.

CNN's were the basis of the Deep Learning boom when [KSH12] defeated its competitors on the ImageNet challenge by successfully applying a Deep CNN for the first time.

2.2 Convolutional Operator

A convolution operation (2.1), denoted with an asterisk $*$, at moment t, is defined as a real valued operation between two functions, f and g, and it expresses how the shape of one function is modified by the other.

$$(f*g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau \tag{2.1}$$

The discrete valued convolution operator is given by Equation (2.2). This formulation of the same operation is necessary because the computational implementation of this operation requires it to support discrete values as input.

$$(f*g)(t) = \sum_{-\infty}^{\infty} f(\tau)g(t-\tau) \tag{2.2}$$

Until now, our convolution formulations have supported kernels which support an infinite real set of values, i.e. $g:\mathbb{R}\rightarrow\mathbb{R}$. When g has a finite support set, i.e. $\tau \rightarrow \{0,1,...,M-1,M\}$, the convolution operation can be described with (2.3).

$$(f*g)(t) = \sum_{\tau=1}^{M} f(\tau)g(t-\tau) \tag{2.3}$$

This is also important since in practice, our input and kernels are arrays of data with limited sizes. Moreover, these inputs and kernels can have multiple dimensions. The convolutional operation can be rewritten to support this kind of multidimensional inputs. A 2D convolution operation is defined by:

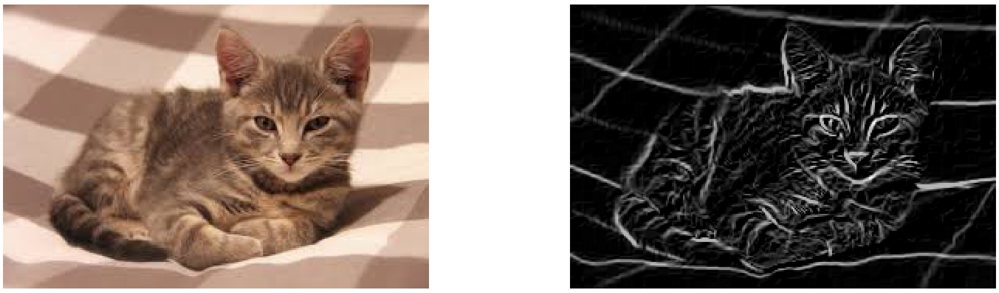
$$(f*g)(i,j) = \sum_{\gamma=1}^{N}\sum_{\tau=1}^{M} f(\tau,\gamma)g(i-\tau,j-\gamma) \tag{2.4}$$

In practice, to avoid the unnecessary complexity of flipping the kernel or the input(convolutions are commutative), we use a very similar operation which is called cross-correlation (2.5), which does not flip neither the kernel or the input. This has no real impact on the operation in convolutional neural networks since the kernel is a set of features to be learned and it does not matter if the kernel is learned flipped or not as long as it is processed in the same manner every time.

$$(f*g)(i,j) = \sum_{\gamma=1}^{N}\sum_{\tau=1}^{M} f(i+\tau,j+\gamma)g(\tau,\gamma) \tag{2.5}$$

An image is the perfect example of a 2D input to a convolution operator as it defined by a grid of pixel values. Convolving the image with a kernel generates a feature map composed of kernel activated features at each block of the image.

In Figure 2.1, it is possible to see an example of the result of a convolution operation.



(a) Original Image.

(b) [Sob14] Kernel convolution result.

Figure 2.1: Example of the application of the Convolution Operation.

2.3 Convolutional Layer

Convolutional layers take advantage of the spatially relevant information of data with a grid-like topology, unlike Fully Connected layers which treats every element, which can be an element of the input or an element of a previous layer which we call a neuron, in the same manner. Each connection in a network contains a weighted parameter and each neuron a bias parameter. These parameters are very commonly referred to as the *weights* of the network. Both of these types of parameters can be updated and are optimized during learning.

A Neural Network which contains Convolutional layers is typically called a Convolutional Neural Network (CNN), although they also usually possess other kind of layers such as Pooling and Fully Connected layers. A CNN is created by stacking these layers, which transform the input in a fully differentiable manner.

A Fully Connected Neural Network is composed by a series of Fully Connected layers. This network receives an 1D input which is iteratively transformed by the Fully Connected layers or hidden layers. "Hidden" comes from the fact that it represents the input as a result of an unknown function and from the fact that when looking at the interface of a network, only the input and output of the network are exposed. As mentioned before, each neuron of a Fully Connected layer is connected to every hidden neuron of the previous layer which creates a global receptive field. However, when dealing with multi dimensional data inputs it becomes unfeasible to connect every hidden neuron to all neurons in the previous layer.

On the other hand, Convolution Layers connect to a reduced local region, in space, of the input, resulting in a local receptive field with the size of the kernel, illustrated in Figure 2.2. The Convolution Layer operation can be seen as a sliding window operation, where the kernel *slides* through the image across its width and height and computes the dot product between the kernel and the image at each location. From this operation results a 2D feature map which shows the activated kernel features at every location in the image. This sliding window operation can be performed multiple times in a single layer if multiple kernels are defined. The stride with which

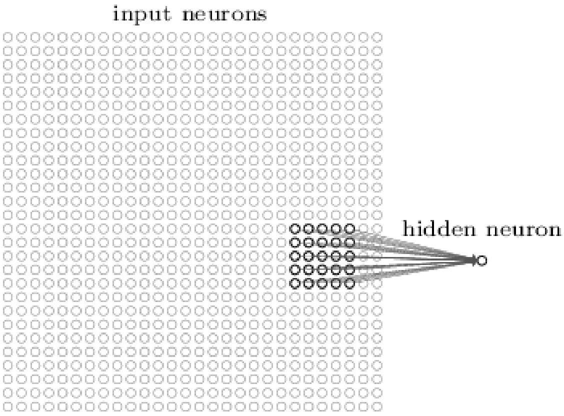


Figure 2.2: Local Receptive Field example taken from [Nie15].

we slide and the number of kernels are hyperparameters that contribute to the shape of the output of a Convolutional Layer.

Since the same kernel is used to slide through the input, the weights of the connections are applied multiple times, therefore the elements of the output share kernel weights. This also form of parameter sharing introduces *translation equivariance* meaning that when the input is translated, the output is translated the same way [GBC16].

By choosing kernels smaller than the input in order to we are able to reduce the amount of connections. This occurs because while a Fully Connected contains $(N+1)\cdot M$ number of learnable parameters, where N and M are the input and output sizes, in Convolutional Layers there are $(N+1)\cdot K$, where $K = W \cdot H \cdot D$ is the kernel size and W, H and D are the width, height and depth of the kernel.

The output of Convolutional Layer operation is described by Equation (2.6).

$$O_{i,j,k} = \sum_{d=1}^{D_I} \sum_{w=1}^{W} \sum_{h=1}^{H} I_{i+w,j+h,d} W_{w,h,d,k} + b_{w,h,d,k} \tag{2.6}$$

2.4 Pooling Layer

Along with Convolutional Layers, Pooling layers are also an important component of a CNN. Usually, Pooling Layers are inserted after a stack of Convolutional Layers. The Pooling operation creates a smaller feature map by applying a filter to subregions of the input. The operation is then applied, much like the convolution operation, in a sliding window like fashion. An illustration of the operation can be seen in Figure 2.3.

Pooling allows the network to become approximately invariant to small translations and at the same time, by reducing the resolution of the feature maps, it helps decrease the complexity of the network. In most applications of a CNN, translation invariance is a useful property since it is usually more important to know if a feature is present than to know precisely where it is. Using Pooling can be seen as adding an infinitely strong prior to the function the layer should learn, dictating it must be invariant to small translations [GBC16].

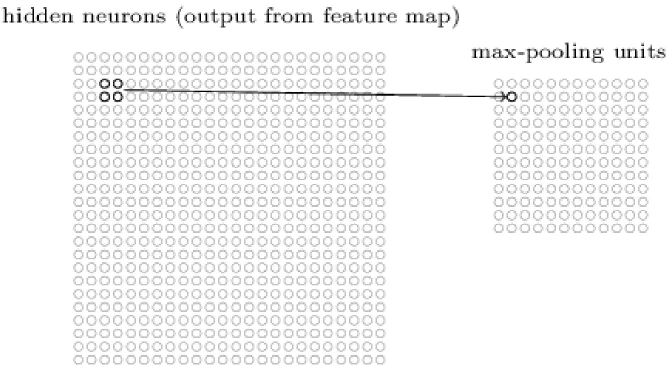


Figure 2.3: MaxPooling illustration example taken from [Kar18].

There are multiple ways to filter a region of features. Here we describe the two most common variations of a Pooling Operation:

- **MaxPooling**: MaxPooling finds the maximum value for each subregion and creates an output where each element is the maximum value of every region.
- **AvgPooling**: AvgPooling averages the feature values of each subregion and creates an output where each element is the average value of every region.

2.5 Activation

The Universal Approximation Theorem [Csá01] states that a neural network single hidden layer can approximate any continuous function on compact subsets of Rn., given enough hidden neurons and an activation function that is non-constant, bounded, and monotonically-increasing continuous, can approximate any continuous function on compact subsets of $\mathbb{R}^n$.

Up until this point, the network would only apply a linear transformation to the input. Moreover, the network would not satisfy the Universal Approximation Theorem therefore it wouldn't be able to learn more complex non-linear functions. In order to do so, we must introduce non-linear activation functions. Activation functions are applied after a linear operation and are an element-wise operation which means it does not affect receptive fields.

Given an activation function σ, the Equation (2.6) is rewritten as:

$$O_{i,j,k} = \sum_{d=1}^{D_I} \sum_{w=1}^{W} \sum_{h=1}^{H} \sigma(I_{i+w,j+h,d} W_{w,h,d,k} + b_{w,h,d,k}) \tag{2.7}$$

The chosen functions need to be differentiable in order for the network to be trained. The most commonly used functions are:

- **Sigmoid**: The Sigmoid function, illustrated in Figure 2.4a, is defined by:

$$sigmoid(x) = \frac{1}{e^{-x}+1} \tag{2.8}$$

This biologically inspired function squashes the input values to a range of $]0,1[$. However, this function has two properties that pose major disadvantages which are the not zero-centered output and the fact it is more prone to gradient saturation.

- **tanh**: The Hyperbolic Tangent (tanh), illustrated in Figure 2.4b, function is defined by:

$$tanh(x) = \frac{e^{2x}-1}{e^{2x}+1} \tag{2.9}$$

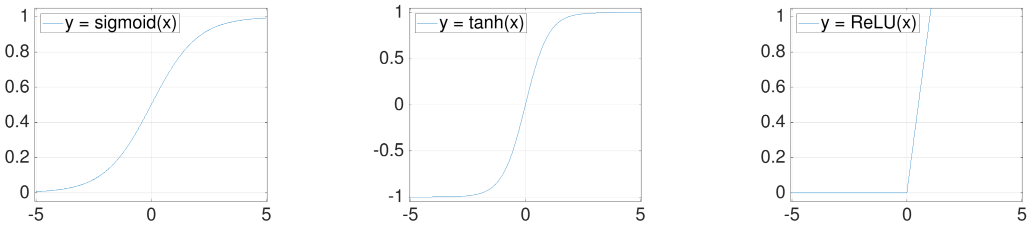
This function squashes the input value to the range $]-1,1[$. Tanh is a rescaled sigmoid function however this version is zero-centered. Nonetheless, it still suffers from the gradient saturation problem.

- **ReLU** [NH10]: The Rectified Linear Unit (ReLU), illustrated in Figure 2.4c, function is defined by:

$$ReLU(x) = max(x,0) \tag{2.10}$$

Along with the reduction in computational complexity, ReLU fixes the gradient saturation issue present in the previous functions however it introduces other problems such as dead ReLU units. Subsequent iterations of ReLU such as LeakyReLU or Exponential Linear Units (ELU) address this issue. This function is not differentiable in all of its domain with its derivative at $x = 0$ being undefined. In order to use back propagation, explained in Section 2.7, every operation in the network must be differentiable. In practice, the derivative of ReLU is defined as:

$$\frac{\partial ReLU(x)}{\partial x} = \begin{cases} 0 & \text{if } x \leq 0 \\ 1 & \text{if } x > 0 \end{cases} \tag{2.11}$$



(a) Sigmoid. (b) Hyperbolic Tangent. (c) Rectified Linear Unit.

Figure 2.4: Illustration of different activation functions.

2.6 Loss

As mentioned before, CNN's usually contain Fully Connected Layers which are generally the last layers to be stacked. After high and low level feature extraction, we can take advantage of the global receptive field of these layers and perform a global transformation of the input.

The last layer is usually one dimensional with as many units as labels in the case of a classification problem or a single unit in case of a regression problem. The values of this last layer either represent the score, in the form of logits, of the input for each class or the estimated regressed value.

In order to train a CNN, we need to come up with an loss or cost function that can measure the quality of model's predictions or regressions, given certain parameters. The network will be optimized in order to minimize this value. The most common loss function for classification problems is the Cross Entropy (2.12), applied after a Softmax (2.13) operation which squashes a vector of arbitrary values to a vector of real values, with each entry having a value in the range $]0,1]$ and $\sum_{k=1}^{K} v_k = 1$, while for regression problems is commonly used the Mean Squared Error (2.14) or the Absolute Squared Error (2.15) (MSE/MAE).

$$\mathscr{L}(\hat{y},y) = -\sum_{n=1}^{N} y_n log(\hat{y}_n) \tag{2.12}$$

$$\sigma(v_k) = \frac{e^{z_j}}{\sum_{n=1}^{N} e^{v_n}} \tag{2.13}$$

$$\mathscr{L}(\hat{y},y) = \sum_{n=1}^{N} (\hat{y}-y)^2 \tag{2.14}$$

$$\mathscr{L}(\hat{y},y) = \sum_{n=0}^{N} |\hat{y}-y| \tag{2.15}$$

2.7 Back Propagation

For the network to output optimal values its parameters have to be adjusted. The optimization technique most commonly applied to CNN's is gradient descent. In order to update the weights we need to compute the gradient of the loss w.r.t. the network's parameters, which are the weights of the connections and bias. Due to the stacking nature of neural networks, the gradient of each layer can be propagated backwards to an earlier layer for a more efficient computation. This algorithm is named Back Propagation and is powered by the use of consecutive applications of the chain rule:

$$\delta(\sigma(x))' = \frac{\partial\delta(\sigma(x))}{\partial x} = \frac{\partial\delta(\sigma(x))}{\partial\sigma(x)} \cdot \frac{\partial\sigma(x)}{\partial x} = \delta'(\sigma(x)) \cdot \sigma'(x) \tag{2.16}$$

The vanilla gradient descent algorithm is defined by Equation (2.17).

$$W^{t+1} = W^t - \lambda \cdot \nabla\mathscr{L}(W^t;\hat{y},y) \tag{2.17}$$

where W^t indicates the model's parameters on iteration t, in this case an epoch, and λ is a weighting parameter called Learning Rate (LR), which dictates the size of the step taken in the direction of the negative gradient.

Up until this point, each parameter update was computed using the entire training dataset. When dealing with simple loss surfaces and small datasets this approach works however in practice we deal with extremely complex and non-convex, smooth loss surfaces and huge datasets in order to fight model overfitting which makes the task of fitting all the data in one iteration unfeasible. Instead, we can divide our training dataset into smaller chunks or batches and iteratively updating the networks parameters using each batch at a time which also takes advantage of adding more variance to the updates since the gradients are noisier. This optimization technique is called mini-batch gradient descent and can be defined by Equation (2.18). The size of the batch can vary and with batch size of one we call it Stochastic Gradient Descent (SGD), defined by Equation (2.19), where W^t now stands for the parameters on the batch iteration t.

$$W^{t+1} = W^t - \lambda \cdot \nabla \frac{1}{B} \sum_{b=0}^{B} \mathscr{L}(W^t; \hat{y_b}, y_b) \tag{2.18}$$

$$W^{t+1} = W^t - \lambda \cdot \nabla \mathscr{L}(W^t; \hat{y_i}, y_i) \tag{2.19}$$

Gradient descent can be further optimize by adding a hyper-parameter momentum, α , a running average of the gradients which helps reduce noisy variations in the direction of the gradient and enhances consistent ones. The update rules are defined as:

$$\begin{aligned} v^{t+1} &= \alpha v^t - \lambda \cdot \nabla \mathscr{L}(W^t; \hat{y}, y) \\ W^{t+1} &= W^t + v^{t+1} \end{aligned} \tag{2.20}$$

Nesterov momentum [SMDH13] improves momentum by looking ahead and calculating the gradient using an approximate weights updated with the current momentum values. The Equation (2.20) can be updated as:

$$\begin{aligned} v^{t+1} &= \alpha v^t - \lambda \cdot \nabla \mathscr{L}(W^t + \alpha v^t; \hat{y}, y) \\ W^{t+1} &= W^t + v^{t+1} \end{aligned} \tag{2.21}$$

Adagrad [DHS11] adapts the Learning Rate hyper-parameter to the model's weights therefore removing the need to tweak the learning rate. It performs large updates for infrequent features and updates for frequent ones. This is done by taking into account the gradient at each update step t and modifying the Learning Rate w.r.t it, where each weight update rule is expressed by:

$$W^{t+1,i} = W^{t,i} - \frac{\lambda}{\sqrt{G_{t,ii}} + \varepsilon} \cdot \nabla \mathscr{L}(W^{t,i}; \hat{y_i}, y_i) \tag{2.22}$$

where $G_{t,i} \in \mathbb{R}^{d,d}$ is a diagonal matrix with each diagonal element representing the sum of the squares of the gradient w.r.t W_i at update step t and with smoothing term ε in order to avoid dividing by zero.

One of the most commonly used optimization techniques is Adaptive Moment Estimation (ADAM) [KB14]. ADAM keeps an exponential decaying average of previous $G_{t,i}$ and of past gradients, similarly to momentum. This information is kept as:

$$\begin{aligned} g^t &= \nabla \mathscr{L}(W^t; \hat{y}, y) \\ v^{t+1} &= \beta_1 v^t + (1-\beta_1) g^t \\ s^{t+1} &= \beta_2 s^t + (1-\beta_2)(g^t)^2 \end{aligned} \tag{2.23}$$

where m_t and s_t are estimates of the first and second central moment of the gradients, which are the expected value of the gradients and the variance, respectively. Since m_t and s_t are initialized with zeros, it is observed that the values are biased towards this value. Adam corrects this estimates by:

$$\begin{aligned} \hat{m}^t &= \frac{m^t}{1-\beta_1} \\ \hat{s}^t &= \frac{s^t}{1-\beta_2} \end{aligned} \tag{2.24}$$

and then updates the weights with Equation (2.25). The hyper parameters β_1 and β_2 are the exponential decay rates for the first and second moment of the gradients, respectively. [KB14] propose default values of $\beta_1 = 0.9$ and $\beta_2 = 0.999$ which they empirically proved to work well in practice.

$$W^{t+1} = W^t - \lambda \frac{\hat{m}^{t+1}}{\sqrt{\hat{s}^{t+1}} + \varepsilon} \tag{2.25}$$

Chapter 3

Literature Review

This Literature Review is divided in 2 sections. The first section is where a summary account of the research done on Age Estimation through 2D facial images is presented. This section is further divided into subtopics which include Appearance Base Feature Extraction and Deep Learning approaches. An additional subtopic on Apparent Age Estimation was added. Although Apparent Age Estimation is not the focus of this project, this research topic raised interesting ideas that might be applied to Biological or Real Age Estimation.

In the second section is available a brief overview of the work that has been done on 3D Face Modelling. This section is subsequently divided into two sections. In the first, the focus is on the study of Facial Landmark Detection while the latter is focused mainly on 3D Face Model Reconstruction.

3.1 Biological or Real Age Estimation

3.1.1 Appearance Base Feature Extraction

One of the first approaches to facial age estimation was done by [HKL94]. In their work, they calculated six different ratios based on distances between facial landmarks. In addition to these ratios, they also implemented a very simple wrinkle detector. At the end, they were able to correctly classify each face from their 47 face image dataset as either baby, adult or senior.

In 2009, [LRBS09] used Active Appearance Model (AAM) [CET01] in conjunction with Support Vector Machine Regression in order to tackle the FG-NET-AD [Lan04] dataset. Principal Component Analysis (PCA) [F.R01] was used to create a statistical model of the face that contains both shape variation and gray-level appearance features. First they modelled the shape variation by applying PCA to hand-annotated facial landmark data. Therefore any example can be annotated by:

$$x = \bar{x} + P_s b_s \tag{3.1}$$

where $\bar{x}$ is the mean shape, P_s a set of orthogonal principal components of the data and b_s a set of parameters. The same is made for appearance features resulting in the following linear model:

$$g = \bar{g} + P_g b_g \tag{3.2}$$

where $\bar{g}$ is the mean shape, P_g a set of orthogonal principal components of the data and b_g a set of parameters. The set of parameters b_s and b_g now define the model. They are further combined by adding a weighting factor and by applying PCA again to remove possible correlations between shape and appearance. To find a model that fits a 2D image, their model minimizes the difference between the original image and the recreated one. The way this is achieved is by learning the relationship between the reconstruction error and the error in the model parameters, therefore a model is trained iteratively to learn how to correct the model parameters according to the reconstruction error.

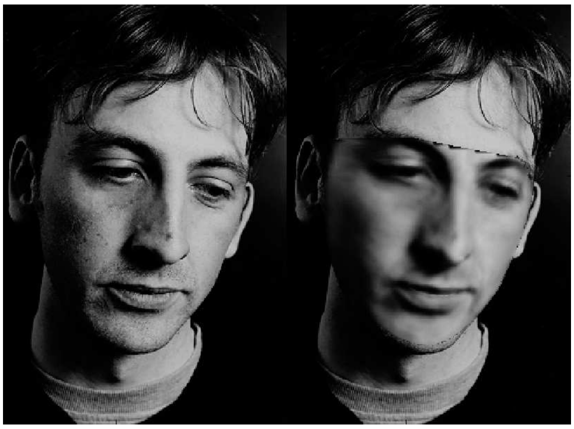


Figure 3.1: Example of an AAM reconstruction (left)/original (right) [CET01].

After the feature extraction process, these features are fed to an SVM [CV95] responsible for binarily classifying the image as an adult or a child. This outcome will be the deciding factor on whether the features are fed to a Support Vector Regression (SVR) model for adults or children. The chosen SVR will be responsible by outputting the final estimated age from a continuous range of age.

[GZSM07] proposed an Ageing Pattern Subspace(AGES) that defined a pattern as a sequence of a single person face images with different ages in temporal order. AGES can generate the missing ages by applying Expectation Maximization based algorithms. When creating a pattern, facial features are extracted using AMM [LRBS09]. The missing ages from the pattern are filled with the mean feature vector from all the individual's facial features. Standard PCA is then applied to all patterns. Then they try to reconstruct the original images using the new pattern projection. The feature vector corresponding to the originally missing ages is updated by the new reconstructed feature vector. A new projection is then created from the new estimated features and the process repeats again until convergence ([GZSM07] provided proof of convergence). The age estimation for an unseen image is achieved by finding the ageing pattern most suitable for the image.

Still within the topic of feature extraction methods based on visual appearance, Local Binary Patterns (LBP) [Ots79] were used by [GN08] as image features since they can represent fundamental properties of the image and its occurrence histogram is an effective texture feature for face descriptions. To perform age estimation on an new face image, spatial LBP histograms are produced and then used to classify the face as one of the six age classes, using a distance based classification method, either by using the distance between the new image's LBP histogram and the mean histogram of each of the age classes or using a K-Nearest Neighbour Classifier [GN08]. Other techniques include using Gabor wavelet [GA09] and using Bio-Inspired Features (BIF), which [GMFH09] improved by using a pyramid of smaller Gabor filters with smaller sizes and problem-specific number of bands and orientations.

3.1.2 Deep Learning

With the increasingly good results Deep Learning techniques continue to provide on the Computer Vision research area, research applied to Real Age Estimation using these techniques has been the focus of a substantial amount of academic work [CZD$^+$17, YLW$^+$14, NZW$^+$16, ZDH17, RTG15, HHM$^+$17, QMB17] over the last years. CNN's are the preferred technique when it comes to feature extraction for images in deep learning. One big advantage of Neural Networks is the ability to transfer their knowledge from one domain to another [OAE16, YCBL14]. Research shows that using pre-trained models on age estimation problems is advantageous [OAE16] whether we use a model trained on a broader classification problem such as the ImageNet dataset [DDS$^+$09] or face specific classification challenge such as with the VGG-Face [PVZ15] model for face identification [QMB17].

[HHM$^+$17] proposed an end-to-end deep embedding network for age estimation. To create a metric embedding we need to learn a function that can map an input to a feature space where the Euclidean distance (3.3) in this embedded space directly represent the semantic similarity of the inputs, and in this case, the features were learnt by triplet loss (3.4) using a CNN. These inputs can be anything from images [SKP15] to words [MCCD13]. They set out to create a multi-task classification problem in which they optimize their model using triplet loss in addiction to age estimation through discrete classification loss. With the addition of the triplet loss, the CNN can learn a metric space where the distance between similar ages is bigger than distant ages. Therefore, the features learnt by the network are more discriminative when it comes to performing the age estimation task.

$$D(I_a, I_b) = ||f(I_a) - f(I_b)||_2^2 \tag{3.3}$$

The triplet loss gains it's name by taking a triplet of inputs (X_t^a, X_t^p, X_t^n) where X_t^a is the anchor of the t^{th} triplet, X_t^p is the positive input with a similar semantic value as the anchor, in this case age, and X_t^n is the negative with a different semantic value. The triplet loss function is represented by:

$$\mathscr{L}(X_t^a, X_t^p, X_t^n, W_T) = max\{0, m + D(X_t^a, X_t^p) - D(X_t^a X_t^n)\} \tag{3.4}$$

where m is the margin parameter and W_T are the weights of the feature extractor for which a CNN was used.

Still on the topic of multi-task classification, [NZW+16] use the age-related ordinal information and propose a multiple output CNN in order to perform age ordinal regression. They transform the ordinal regression into a series of binary classification sub-problems. These binary classifiers are responsible for predicting whether the rank of an image is above a r_k rank, $k \in K$, where K is the number of ranks, i.e. number of age ranges, which can be discrete or aggregated values. This technique was later improved by [CZD+17].

Other similar approaches were developed by [YLL15] and [TCT16], where the multi-task classification loss is formulated by adding the gender and gender specific age estimation losses. Multi-task learning can be seen as a form of inductive transfer which can help improve a model by introducing inductive bias. The inductive bias in the case of multi-task learning is produced by the sheer existence of multiple tasks, which causes the model to prefer the hypothesis that can solve more than one task. Multi-task learning usually leads to better generalization [Rud17].

More recently, a lot of focus on data augmentation using Generative Adversarial Networks [GPM+14] has been popping up in literature in many fields, such as autonomous driving [ASE17], medical imaging [FKA+18a] and also face identification [ZXKJ+17]. The idea of a GAN is to create a system where two networks are in constant competition in a zero-sum game. While one network tries to generate synthetic data indistinguishable from the original source data, the other is responsible for distinguishing generated data from original data. [FKA+18b] took this idea and applied it to the age estimation task. It took both FG-NET-AD and MORPH2 datasets and built a synthetic augmented instances of faces based on a simple trained GAN.

The current state of the art performance is hold by [ZLY+18] on both MORPH2 and FG-NET-AD datasets. They extract local facial characteristics from a cropped region estimated by a Long-Short Term Memory (LSTM) unit. They then combine their local extracted features with the global-image level features to perform their final estimation.

An overview of performance on datasets MORPH2 and FG-NET-AD can be seen in Table 3.1.

Table 3.1: Error in Mean Absolute Error (MAE) on datasets MORPH2 and FG-NET-AD.

	MORPH2 Dataset	**FG-NET Ageing Database**
AGES [GZSM07]	8.83	6.77
Bio-Inspired Features [GMFH09]	-	4.77
OHRank [CCH11]	6.07	4.48
CNN(Multi-Task) [TCT16]	3.63	-
OR-CNN [NZW+16]	3.27	-
DEX [RTG15]	3.25	4.63
Ranking CNN [CZD+17]	2.96	-
Deep Embedding [HHM+17]	2.71	3.19
DEX(IMDB-WIKI) [RTG15]	2.68	3.09
DLDL [GXX+16]	2.42	-
AL-RoR [ZLY+18]	2.36	2.39

A summary overview of the research done on age estimation using a single image utilizing the FG-NET-AD [Lan04] has been made available by [PLTC16]. It shows a consistent increase in interest on facial age estimation between 2005 and 2012.

3.1.3 Apparent Age Estimation From a Single Images

In 2015 and 2016, ChaLearn ran the Looking at People (LAP) apparent age estimation from a single face image challenges [EFP+15]. These challenges differ from traditional real age estimation in the sense that instead of focusing on trying to predict the biological age from images, it focuses on estimating age as it is perceived by other human beings. Along with the challenge proposal, ChaLearn made available the biggest datasets of apparent age annotated face images, the LAP datasets, one for each year of the competition.

In order to successfully tackle the 2015 challenge, [RTG15] from the Computer Vision Lab, ETH Zurich, proposed an ensemble of Convolution Neural Networks, using the VGG16 [SZ14] pretrained on the ImageNet dataset for image classification [RTG15], approaching the problem as multi-class classification problem and ended up winning the competition. They finetuned the ImageNet pretrained VGG16 network using their own collected dataset they called IMBD-WIKI. The dataset is publicly available at their website and contains 524,230 images. These images were then processed in order to obtain an aligned cropped image of the face by using an iterative rotation process without landmark detection. Afterwards, they divide the LAP dataset in 20 subsets, using each one to further finetune their IMDB-WIKI finetuned VGG16 network, therefore creating 20 different networks. When trained for classification, the output layer is composed of 101 softmax-normalized neurons, representing discrete ages from 0 to 100. They improve their predicted output by computing a softmax expected value (3.5):

$$E(O) = \sum_{i=0}^{Y} y_i o_i \tag{3.5}$$

where O is a random variable with a Y number of finite outcomes, in this case 101, from 0 to 100, $o_i \in O$ and y is the softmax output of the network for a given y. The final prediction is made by averaging every networks prediction.

In 2016, [MAE16] got fifth place by making very good use of an ensemble of VGG-Face [PVZ15] networks finetuned with the IMDB-WIKI dataset [RTG15]. Instead of representing every discrete age as a class, they use a softer age encoding and distributed the ages in groups of three to try and reduce the impact of the existing standard deviation of apparent age on the LAP dataset. After training and finetuning their models individual with a subset of augmented images from the LAP dataset, they pick the top 5 predicted classes of each model to do a weighted average which is then averaged for all 3 models to produce the final output.

[ABBD16], the winners of the 2015 LAP Challenge [EFP$^+$15], noticed that the 2016 LAP dataset, in contrast to the 2015's dataset, contained a considerable amount of face images of children in comparison with the previous dataset iteration. The standard deviation of the perceived age of children according to the dataset is inferior to the average over the whole dataset, which meant a bigger penalty would occur for each misclassification of a child's apparent age. With this fact in mind, the authors created a 2-step approach using ensembles of VGG-Face [PVZ15] finetuned with the IMDB-WIKI dataset [RTG15]. The authors finetuned 11 networks separately with the competition dataset, using, instead of a discrete one hot encoding, a normalized label distribution according to the image face standard deviation, which they called the "general" prediction. Separately, 3 VGG-Face [PVZ15] CNN's were used as pretrained models to classify children's age using a private dataset containing only children face images. These networks were trained using a one hot encoding. The two ensembles were then combined in such a way that if the "general" ensemble would classify an image as containing the face of a child, it would delegate further classification to the ensemble specialized in children.

3.2 Face Modelling

3.2.1 Facial Landmark Detection

An Active Shape Model (ASM) [CTCG95] is a model-based approach to modelling objects whose appearance is flexible. A deformable parametrized model shape is produced based on a training set and its parameters learn how the object's shape can change. Therefore, it is able to fit new instances with similar shape. This involves an iterative procedure that requires following the new image's gradient in order to fit a given model. [LTC97] applied this idea to facial shapes. This work allowed them to fit a face to a trained model using a 2D image. The resulting fitted model expresses the main facial keypoint locations as well as its shape. This approximation can then be used for gender classification, expression recognition, 3D face recovery, age estimation and face alignment. AAM [CET01] extends this idea by combining the shape model with another parametrize model, the later representing the facial texture.

One of the first approaches to facial landmark detection making use of deep learning was developed by [SWT13]. The idea behind their work involved the use of a cascade of Convolutional Neural Networks. Distributed by 3 levels, the use of multiple stacked and parallel CNN allowed for a coarse-to-fine prediction, which involved feeding level 2 and 3 of the network with a cropped version of the original image around an initial facial landmark estimation, resulted in a more robust and accurate estimation of the points' position. The project described by [SWT13] is able to regress the position of five facial landmarks.

Due to an increasing amount of lack of correspondence between datasets, in 2013 [STZP13] proposed a semi-automatic methodology as well as defining an academic standard for facial landmark annotation, based on MultiPIE dataset annotation [GMC$^+$08], with 68 well defined points of interest, the 68 iBug facial point annotations. Their tool allowed researchers to create a less error-prone way of annotating, that encourages cross-database experiments and have a substantiation amount of high quality landmark features. Building on top of the idea of AAM [LRBS09], [STZP13] created Active Orientation Models (AOM), proposing a model capable of outputting the 68 landmarks given a 2D image. They take an annotated subset of images with which they train an AOM. They then feed the AOM with a non-annotated subset. If the output is not manually accepted as "good", another AOM is trained using both the original annotated subset and the "good" annotations of the previous AOM and another batch of synthetically produced annotations is created. When all images are accepted, the annotations on the non-annotated subset are manually checked and corrected. This tool allowed them to produce annotated datasets with the same model from several previously non-annotated databases which they consider so accurate that they can be used as ground truth.

[BT17b] proposed a network which converts 2D landmarks annotations and converts them into 3D, allowing them to create the biggest 3D facial landmark dataset to date (of around 230 thousand images). They stacked 4 Hour-Glass networks [NYD16] on which they replace the Hour-Glass's bottleneck residual block [HZRS15a] with [BT17a] version, which was shown to outperform [NYD16] when the same network parameters were used. They added to the common 3 channel input (RGB) 68 additional channels, each representing a 2D landmark with a 2D Gaussian distribution with a unit standard distribution around the landmark's location. The network is able to then estimate the depth of 2D projections of the 3D landmark position using a Res-Net-152 [HZRS15a] network adapted to accept the image and the 2D projections as input and output the depth of each landmark.

3.2.2 3D Face Model Reconstruction

For many years, researchers have been trying to find a way to efficiently model a human face in three dimensions. One of the first to tackle this challenge was [Par74] in his PhD dissertation in 1974. He built a parametric model with each specific feature of the face being individually designed and defined. By changing each face feature's specific parameters he could generate different facial shapes and expressions.

In 1990, [BV99] proposed 3D Morphable Model (3DMM), the technique that would later become the most used approach to 3D facial shape reconstruction. They define a morphable face model as a 3D triangulated mesh built from a set of scanned faces. To create their model, they take each face from their set, and represent its geometry with a shape-vector which contains the 3 dimensional coordinates of each vertex of the mesh and a texture-vector, which in turn represent the RGB color values of each correspondent vertex in the shape-vector. By tweaking each element in the shape and texture vector, it is possible to generate new shapes and textures. Assuming all faces are in full correspondence, i.e., geometry vectors follow the same vertex order for every face, they fitted a multivariate normal distribution to their dataset containing 200 facial scans of young adults (100 female and 100 man), based on the averages of the shape and texture vectors. Then PCA [F.R01] is employed to construct a model with reduced number of parameters and these parameters are transformed into uncorrelated variables. They state they maintain 95% of the meshes variance by keeping the 100 principal components of each vector.

Arbitrarily, new face models can then be generated by varying the set of coefficients. However it is necessary to quantify the results in terms of their plausibility of being faces. The fitting of a new 2D face image to a 3DMM model is possible by first pre-aligning the face with the model and then updating the model parameters as well as rendering parameters by applying gradient descent optimization to the recreation error. The recreation error can be defined as the difference between the 2D image and the reconstruction of the image using the rendering of the face using the current 3DMM parameters as well as the rendering parameters. This method was then augmented to allow for extra parameterization of facial expressions [CWZ+14].

A combined use of the 3DMM [BV99] with 2D AAM [CET01] was shown to work by adding additional parameters to the AAM's shape model in order for it to be able to represent 3D poses. Although they find that by constraining the AAM parametrization with a 3D shape model leads to an increase of the number of parameters up to a factor of 6, the new model possesses reduced flexibility which in turn leads to faster convergence.

A team at Imperial College London recently released their Large Scale Facial Model (LSFM) [BRZ+16]. This model is constructed out of more than nine thousand 3D facial scans. This allows for a better generalization of the face model in comparison with the 200 hundred facial scans, whose subjects have similar ethnicity and age, used by [BV99]. Due to their dataset's diversity, they successfully created a collection of models tailored by gender, age and ethnicity. They also found a correlation between the 3DMM parameters and the age of the individual, reinforcing the statement that the 3D facial elements do contain pertinent age information.

Instead of fitting a 3DMM in an expensive iterative process that requires the rendering of the model at each iteration of the refinement process, [THMM16] designed a CNN capable of performing 3DMM parameter estimation directly from an input image. This allowed them to create a system with not only with a lower model reconstruction error but also a much faster 3DMM parameter estimation as a result of the rendering bypassing. Later on, they revamped their model in order to adopt expression [CTH+18] parameters and more recently 3D reconstruction under extreme conditions, such as occlusions [THM+17].

[JBAT17] were able to bypass both the face alignment phase and 3DMM fitting by demonstrating that a single volumetric regression of the 3D face geometry is possible using only a 2D image. This reconstruction is made possible by developing an end-to-end architecture that uses two Hour-Glass network [NYD16] with skip connections and residual blocks [HZRS15a]. They also contribute by extending their architecture to a multi-task one where they fork their first HG into an HG responsible for regressing the 3D face structure and the second for extracting the 68 iBug facial landmarks [STZP13] in the from of a 2D Gaussian per channel corresponding to each landmark. [JBAT17] also added a guided network, instead of forking their tasks, because they argue that using facial landmarks can improve the network performance either during training or inference.

In early 2018, [FWS+18] proposed U.V. Position Map as the representation of a full 3D facial structure. Their U.V. coordinate system was created based on 3DMM [BV99], however the resulting regressed model is not constrained by the original 3DMM model. The structure of the U.V. map can be seen in Figure 3.2.

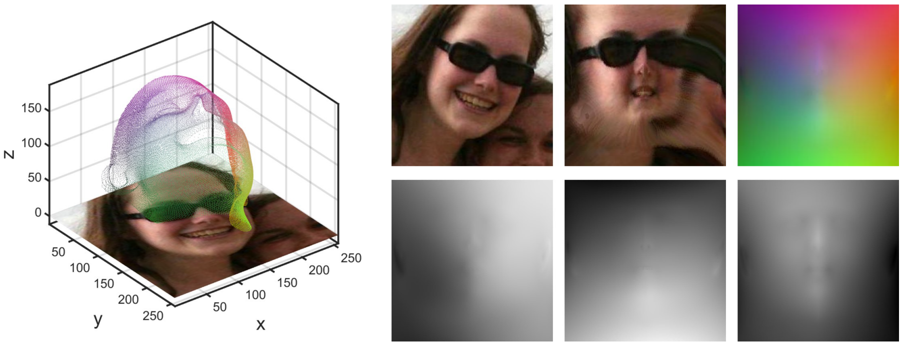


Figure 3.2: Structure of U.V. mapping [FWS+18].

This way, by directly regressing the position map from 2D images they can obtain both the 3D facial structure as well as the resulting face alignment. They train an encoder-decoder network to learn how to transfer the 2D RGB image into a regressed position map. In order for their model to learn its parameters, a weighted MAE (3.6) between the regressed position map and the ground truth position map is used, where M is a weighting mask. The weighted component arises from the fact that MAE treats every point in the position map equally although some regions of the face contain more discriminative features. Therefore, their weight map helps enhance learning in regions corresponding to the centre of the face as well as the 68 iBug [STZP13] landmarks while neglecting the impact of regions such as the neck and body.

$$Loss(\theta) = ||P(x,y) - \hat{P}(x,y)|| \cdot M(x,y) \tag{3.6}$$

They outperform [BT17b] on the 3D landmark alignment task mainly due to the avoidance of using an extra network to estimate the depth position of the landmarks. As a result of this, it is also able to reduce the runtime by more than 5 times when compared to [BT17b] as well as the model size, from 1.5GB to 160MB. [FWS+18] have made their code and pre-trained models available on Github, although they don't share their training details.

Chapter 4

Methodology

4.1 Datasets

In this section, we will explore some of the existing publicly available datasets for facial age estimation. We will investigate their origin, image quality, size and age range available. We will also evaluate their value in relation to each other, describing the weaknesses and strengths of each dataset.

4.1.1 IMDB-WIKI

The competition winners of the 2015 Looking at People International Conference on Computer Vision (ICCV) Challenge [EFP$^+$15] scraped the internet, or more precisely, the Internet Movie Database (IMDB) website and Wikipedia's biographical pages for images of faces with the its corresponding age information, creating the largest publicly available dataset of face images with age labelling, entitled the IMDB-WIKI dataset [RTG15].

Web Scrapping refers to accessing web pages, finding specified elements of that page, extract them and collect said extracted data in structured dataset [BW16]. In order to collect the image data with the correct age labelling, [RTG15], a team at ETH Zurich, extracted only the images which possessed timestamps (the date and time the picture was taken) in its metadata, a human face and the correspondent individual's date of birth (as well as gender, which is not pertinent to this dissertation). They collected a total of 523,051 images, 460,723 and 62,328 from IMDB and Wikipedia respectively. Along with the dataset, they released a file containing all the meta information regarding the images such as: date of birth of the person in the image, data of image capture, gender, name, coordinate based location of the face in the image and face score. With this information, they were able to calculate the age of the person in the picture at the time of the picture's capture and label the image accordingly.

The IMDB-WIKI dataset, although the largest available, does bears some a few cons. Since images are gathered by IMDB from many sources, images have a wide range of resolutions, fields of view and size ratios. Moreover, the dataset was not manually analysed, therefore any error from the web scrapping pipeline is not detected. This errors come in the form of wrong labelling, wrong timestamps, mostly due to the nature of the images. Most images retrieved from the IMDB website are still frames captured from movies, which can have long production times making the timestamps and labelling incorrect.

The full dataset is available at the ETH Zurich Computer Vision Lab's website, including cropped and raw versions, as well as pretrained VGG-16 Caffe models.

As mentioned in Section 4.1.1, the IMDB-WIKI dataset is an extremely noisy dataset due to its content's origin. The collected data contains low quality, non-human facial images such as sketches and animated figures, blank images, wrongly labelled images, among other issues. For this reason, [ZGG+17] spent a week manually removing non standard images. The resulting clean dataset was called IMDBWIKI-101 and possessed 440,607 out of the original 523,051, which means about 16% of the data was considered noisy. However, the team did not make their dataset available to the public, therefore we do not make use of this dataset.

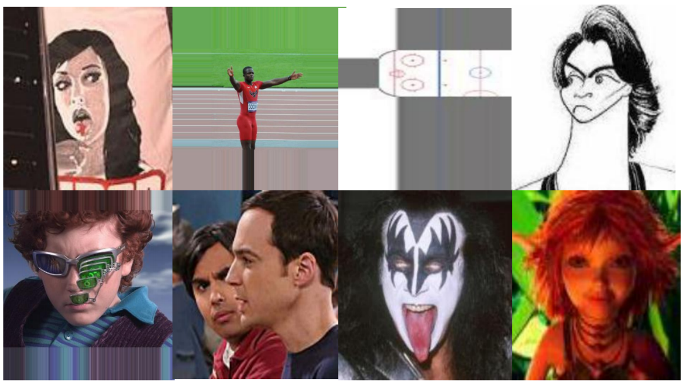


Figure 4.1: Example of non-standard images on IMDB-WIKI dataset [ZGG+17].

4.1.2 MORPH2

The MORPH2 album release for Non-Commercial uses contains 55,000 unique face images of little over 13,000 individuals and it is the largest longitudinal face database. It is also the largest manually inspected dataset for age estimation studies. The images were taken in a controlled environment from 2003 to late 2007. The ages present on the dataset range from 16 to 77. The average number of pictures per individual is 4 with an average of 164 days between each picture. More details are available in the MORPH Non-Commercial Release Whitepaper. Each image of the dataset has a resolution of either 200x240 or 400x480, maintaining the image ration whatever the resolution.

The MORPH Database for academic use was available for free until 2017 and now costs 199.00$ US dollars. However, we directly contacted Dr. Karl Ricanek Jr., director of the Face Ageing Group, who was able to provide us with the dataset for free. We would like to thank Dr. Karl Ricanek Jr. for his generosity and support. The dataset is now available at the Face Ageing Group's website.

4.1.3 FG-NET Ageing Database

The FG-NET Ageing Database, or FG-NET-AD, was created in 2004 by the Face & Gesture Recognition Working Group, a multi institution initiative funded by the European Union. The aim of the project was to encourage research and the development of technologies in the area of face recognition and age estimation [Lan04].

This dataset contains 1002 face images of 82 different people, with their ages ranging between 0 to 69 years. The dataset is organized to present images of the same person in several stages of their lives. The pictures were not taken in a controlled environment and some, specially in the range from 0-10, are grayscaled. The dataset does not have a fixed image resolution. In addition to this, the images also contain a wide range of resolutions, image sharpnesses, scene illuminations, 3D poses and facial expressions. [PLTFC15] gives a more detailed overview of the dataset, including an overview of the research done in several areas with a reliance on the FG-NET-AD dataset.

The original website supporting the free release of the FG-NET-AD dataset is not online at the time of writing this thesis. Alternative mirrors are available elsewhere. We downloaded our dataset from Yanwei Fu's personal website.

4.2 Data Pre-Processing

In this section, we will summarize the dataset pre-processing pipeline of our implementation. Each datasets goes through an iterative transformation process before all the data is extracted from said dataset.

4.2.1 Dataset Oriented Pre-Processing

Each dataset has particular characteristics that force us to approach its processing in a different way. This might be because of file format, labelling technique, folder organization, image characteristics (resolution, aspect ratio, colour encoding), among others. Below we will go over aspects of data pre processing required for each dataset.

The creators of the IMDB-WIKI used an iterative process of rotating the image, running the off-the-shelf face detector of [MBPVG14] and picking the detection with highest detection score. Afterwards, they rotate their cropping with a 40% margin in order to obtain an up-frontal position [RTG15]. They state that in their initial observations, this method works better than landmark detection then alignment and for this reason they adopt this strategy.

In this thesis, we use their publicly released cropped images. These images are cropped using the strategy previously described. However, we make use of the available provided image metadata and only use images that obtained a face score of 1.0, the maximum value. Moreover, we discard any image where a second person is detected (which increases the risk of wrong age labelling) as well as any image with a labelled age above 100 years old. The reason we are imposing very strict rules, is because we want to be assured high quality images even though there is no previous human validation. After all this processing, we are left with over 100 thousand images which makes it a too large of a dataset to perform the next steps in a reasonable time with the resources we have. For this reason, we randomly sample 30 thousand images from the high quality set. We should point out that even after such a severe sampling, the subset is still larger than the FG-NET-AD and MORPH2 datasets.

After this initial processing, IMDB-WIKI images follows the same process as FG-NET-AD and MORPH2. Both datasets have their images already cropped around the face of the person since the images were manually inspected. Therefore, it is safe to assume that every image in these datasets have a single face present in the image.

Although the age labelling on the FG-NET-AD and MORPH2 datasets is contained within each image filename they do not employ the same format. Therefore, we relabel the images by the standard of having the last 3 characters before the file format be the person's age, from "000" to "100".

Be that as it may, we want our results to be comparable to the available literature benchmarks therefore we must adhere by some previously set standards. In 2011, [CCH11] manually selected 5,492 images of people of Caucasian descent, out of the 55,608 available images on the MORPH2 dataset. The reason behind this selection was to reduce the variation between different ethnic groups present in the dataset. Since then, most literature follows this setup [RTG15, HHM+17, NZW+16, WGK15, ZLY+18] and to be able to perform a fair comparison we also adopt this setup.

4.2.2 Training/Validation/Testing Dataset Splitting

To validate the results of our models and ensure the models are capable of generalize to previously unseen data we need to split the model in a training and testing set. We also create subset of the training set to be used as a validation. Validation data will be used to estimate how well our model is generalizing and performing reductions to the algorithm's learning rate or perform Early Stopping if the model either ends up overfitting the training data or we fail to see significant improvements each epoch.

Even though we wish for our results to be comparable to literature, we will not follow the standard validation process for the FG-NET-AD dataset. Over the years, researchers would use Leave One Out(LOO) cross-validation to validate their results on this dataset. This implies creating more than 82 identical models, one for each person, and trained them with the dataset while leaving one person out and each person being left out once. This is not feasible in the scope of this thesis due to complexity of the models used, time constraints and available computational resources. Instead, we will employ the same validation strategy we use for MORPH2 and IMDB-WIKI.

We follow a 60/20/20 percentage split methodology on our training, validation and test sets. Although there is no standard when it comes to the splitting percentage, our approach has been previously used in literature [RTG15, HHM+17, GXX+16] and proven to give good results. More importantly, each set should accurately represent the overall data distribution of the dataset. We randomly split the original dataset, we should expect every split to follow the same distribution.

We confirm this to be the case in Figure 4.2, 4.3 and 4.4 with the FG-NET-AD, MORPH2 and IMDB-WIKI age distributions histograms.

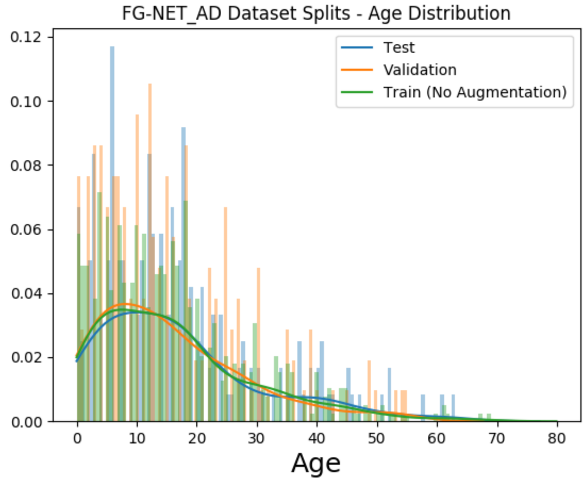


Figure 4.2: Age distribution on train/test/validation split of the FG-NET-AD dataset.

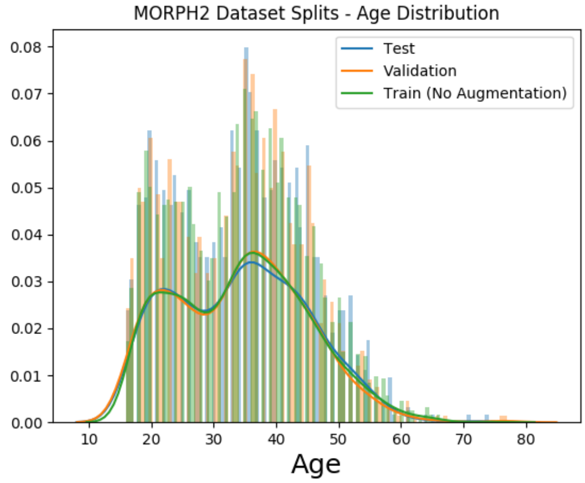


Figure 4.3: Age distribution on train/test/validation split of the MORPH2 dataset.

4.2.3 Data Augmentation

Deep learning techniques are very prone to overfit if fed with a small amount of data, i.e. the models end up memorizing the training data, becoming incapable of generalizing to new unseen data. One of the deciding factors for the success of the AlexNet [KSH12] model in 2012 was the fact they used extensive data augmentation techniques on the existing ImageNet dataset. By using a combination of cropping and flipping techniques they were able to augment their dataset by a factor of 2048. Without it their model suffered from "substantial overfitting" and they would

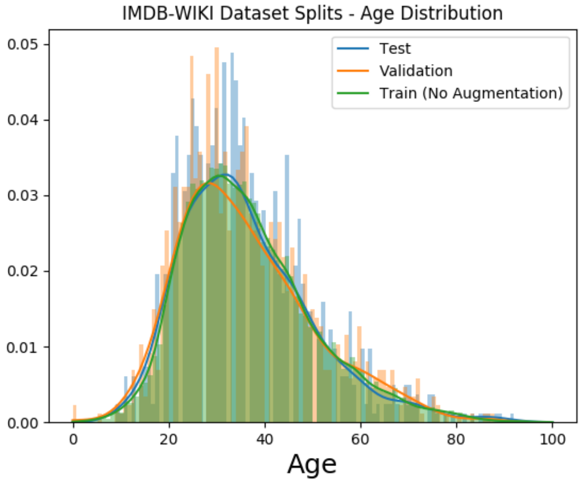


Figure 4.4: Age distribution on train/test/validation split of the IMDB-WIKI dataset.

have been forced to use a smaller model. With this fact in mind, we will employ data augmentation techniques on the used datasets, except on IMDB-WIKI. This decision is due to the fact that IMDB-WIKI is large enough to be used without any augmentation. Moreover, the results comparisons with state of the art approaches will not include IMDB-WIKI because no information about the performance of age estimation models has ever been reported on this dataset. Therefore there is no need to extensively test the performance of the models on this dataset. These factors, coupled with limited computational power availability, support this decision.

Different types of data augmentation in the image field are available. We are now going to summarize some of the possible image transformations:

- **Flipping**: Flipping images creates a mirror image of the original image. The flipping axis can either be horizontal or vertical. It is computationally efficient since it only requires to invert either rows or columns of the image. In our implementation we only perform a horizontal flip [RTG15, KSH12, BT17b]. Vertical flipping is not useful within this context since faces are usually captured in an upwards position and making the model robust to this kind of data would be inconsequential.

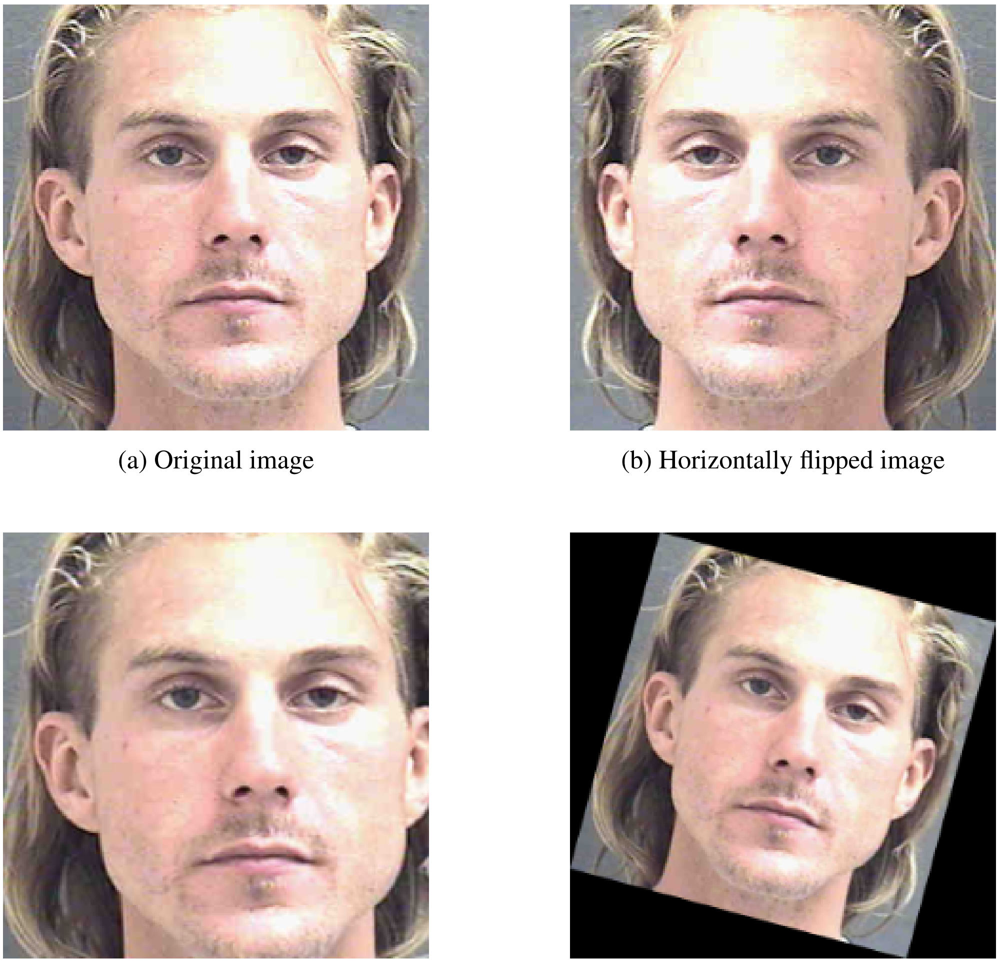
- **Cropping**: Cropping relies on sampling a sub-region of the original image. In our implementation we perform 10 random crops of a region 80% smaller of our original image and its horizontally flipped version. [RTG15, KSH12]

- **Rotation**: Applying a rotation to an image might results in it having different bounds. This means the image will then have to be rectified. In our implementation, we rotate our preprocessed images by -15 and 15 degrees. We then rectify our rotated images by fitting, through downscaling, the rotated image in the same input space as the original image and zero pad the remaining space, which means the original image will be scaled down in addition to being rotated.[RTG15, KSH12, BT17b]

- **Channel Shifts**: This transformation encompasses the addition of random distortions to all pixels in a pre-determined colour channel. We do not implement this transformation in our implementation.

- **Noise Addition**: Adding pixel-wise noise to images can simulate acquired noise from real time cheap systems. Salt and Pepper noise (or impulse noise) can occur when sharp errors occur in analog-to digital converters. Gaussian noise (or Gaussian white noise) can be generated during the capture of a digital image due to poor illumination, high temperature or transmission errors. Every pixel of each image has random noise added to it, sampled from a Normal distribution with $\mu = 0$ and $\sigma = 5$.

An example of the effects of our data augmentation transformations on the MORPH2 train split can be seen in Figure 4.5.



(a) Original image

(b) Horizontally flipped image

(c) Cropped image

(d) Rotated image

Figure 4.5: Several Transformations of a MORPH2 image (before noise addition).

We chose to implement Flipping, Rotation and Cropping because these transformations are easy to implement, computational efficient and allow us to increase the size of our datasets by a factor of 20. We can also add that these data transformations result in the largest increases, according to [TN17], which details several experiments with a multitude of data augmentation techniques on the Caltech101 dataset [LAR03], validated with a 4-fold cross validation.

It is also important to point out that we only perform augmentation techniques on training data. We augment 602, 3,296 and 30,000 images from the FG-NET-AD, MORPH2 and IMDB-WIKI datasets, respectively, while leaving 400, 2,196 and 2,000 images for validation and testing purposes which are then split evenly between both subsets. Validation and testing data are kept untouched after the face detection and cropping section, in accordance to other research studies which will allows us to have comparable results.

4.3 3D Facial Features

There are multiple datasets on 3D face recognition, animation and reconstruction such as BU-3DFE [YWS+06], Bosphorus 3D Face Database [SAD+08], Florence 2D/3D Face Dataset[BBM12], CASIA 3D [XWTQ04], 3D-TEC [VBF11], among others. However due to the fact that neither of them was created in order to tackle the age estimation task, data is always inadequate because it either lacks diversity, as we need samples of different people in order to generalize to everyone, age range, as we would want to sample uniformly all possible ages, quantity, as we would like the overall number of scans to have a significant size and the scan's respective age labelling because most times it is not given. More detailed information can be seen on Table 4.1. Another key problem is the lack of age estimation related studies on this datasets, making our results non-comparable to currently existing research.

Table 4.1: Detailed information on 3D Face Datasets.

Dataset	No. Scans	No. Different People	Labelled Age	Age Range
BU-3DFE	2,500	100	No	18-70
Bosphorus	4666	105	Yes	25-35
Florence 2D/3D	212	53	Yes	22-61
CASIA 3D	4624	123	No	N/A
3D-TEC	428	214	No	N/A

For this reason, our research calls for an extension of the exiting 2D datasets with 3D extracted information in order to make 2D and 3D age estimation domains comparable. A crucial aspect of this research is the effectiveness of detecting facial landmarks exclusively by analysing the provided 2D image. For this reason, we make sure we are able to detect the 68 iBUG [STZP13] landmarks using the DLIB library [Kin09]. DLIB's facial landmark detection makes use of a trained Histogram of Gradients with a combined linear classifier, an image pyramid and sliding window detection scheme [KS14]. The landmark locator was trained using the iBUG 300-W face landmark dataset [SAG+16]. This data validation step is specially important to perform on the augmented training set since the data augmentation techniques might have altered the image in such a way that it becomes impossible to detect a face on the post-processed image. We do not realign the images since that would undo the cropping augmentation step.

After removal of unwanted samples our final distributions of the dataset splits of FG-NET-AD, MORPH2 and IMDB-WIKI can be seen in Figure 4.6, 4.7 and 4.8, where is possible to confirm that the distributions remain the same. The plotted line on this Figures represents a multivariate kernel density estimation which can be interpreted as a generalized histogram density estimation.

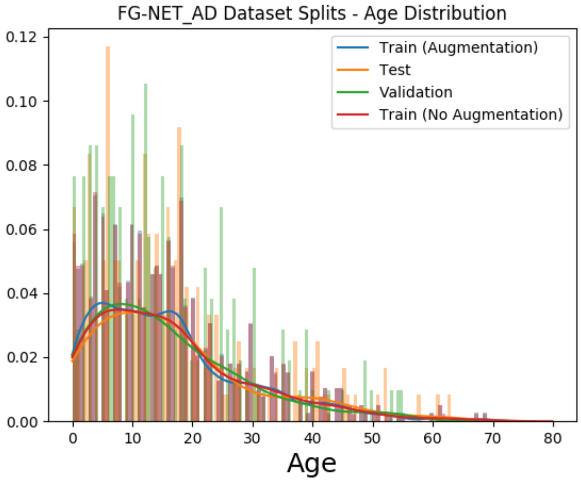


Figure 4.6: Age distribution on train/test/validation split of the FG-NET-AD dataset after data augmentation.

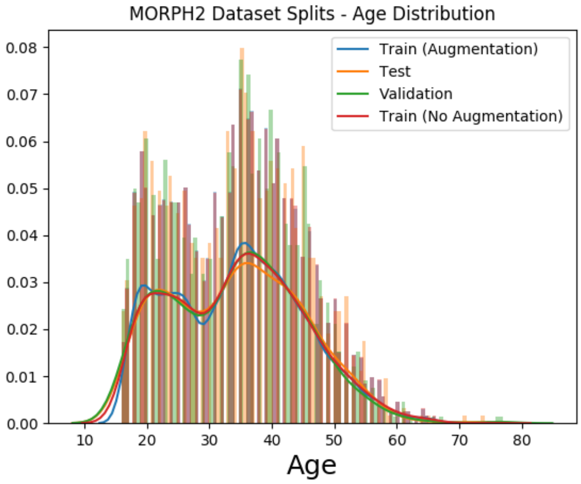


Figure 4.7: Age distribution on train/test/validation split of the MORPH2 dataset after data augmentation.

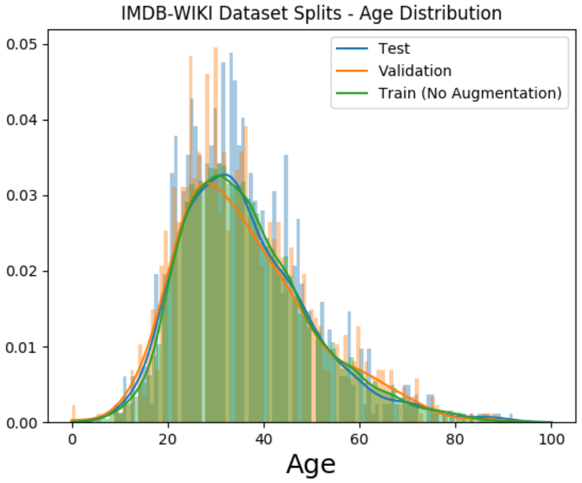


Figure 4.8: Age distribution on train/test/validation split of the IMDB-WIKI dataset after data augmentation.

4.3.1 Landmarks

Facial landmarks are one of the most important descriptors of a face. Landmarks are extensively used in expression classification and 3D reconstruction. Using landmark position as an age estimation enhancement serves as good baseline of what is possible. However, none of the previously mentioned dataset have facial landmark information. For this reason, we set out to create a dataset of 2D facial images with correspondent landmarks and age labelling. Seeing as we lack the manpower and time to manually label every image of our datasets (and respective augmented images), we synthetically label our dataset using current state of the art facial landmark estimators. Note that the implementation of our own landmark detection system is out of the scope of this thesis. We will use the public available code of [FWS+18], which we already covered in Section 3.2.2. At the time of writing this thesis, [FWS+18] hold the current state of the art on iBug [STZP13] 3D landmark alignment and 3D face model regression. The source code is available on Github along with pre-trained models. We generated the 68 iBug landmarks for every image in the augmented, non-augmented, validation and testing split. An example of the 68 iBug landmarks can be seen in Figure 4.9.

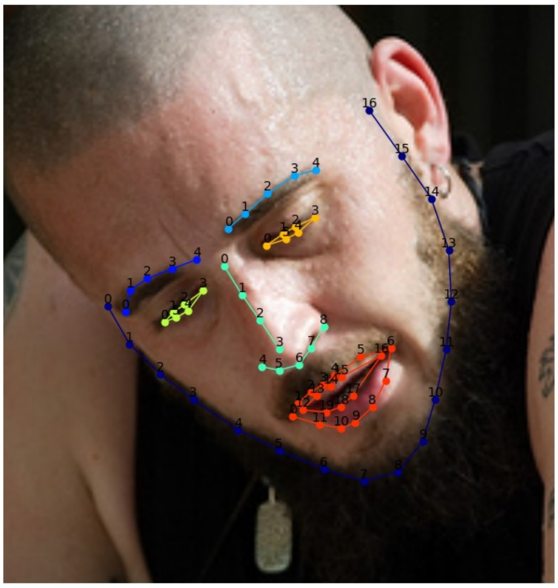


Figure 4.9: Example of the 68 iBug Landmark notation [STZP13].

Moreover, this implementation allows us to not only gather the 2D facial landmark as well as their respective position in regards to the Z axis, i.e. we can estimate the 3D position of the landmark.

4.3.2 3DDM Model Fitting

As we did before where we create a synthetically landmark annotated age estimation dataset, we also set out to fit every facial image in our datasets to a 3DMM model. We want to record the parameters used to fit the 3DMM default model to each face image. As mentioned in section 3, a 3DMM is a statistical model of a 3D representation of an object. The original application of this process and many subsequent works fitted the parametric model by iteratively rendering the model with the current parameters and comparing the rendered image with the input image [BV99]. Now more advance techniques using CNN's are used to directly estimate the 3DMM parameters [THMM16], bypassing the iterative process of rendering the model and comparing it to the original image.

Multiple facial 3DMM are available such as the 2009 Basel Face Model [GFB$^+$17], as well as its more recent update in 2017, the Surrey Face Model [HHT$^+$16] from the Centre for Vision, Speech and Signal Processing of the University of Surrey and the Large Scale Facial Model [BRZ$^+$16] from the iBug group at Imperial College London, which unfortunately is not available for researching purposes outside the medical field.

Making use of Patrick Huber's EOS 3DMM fitting tool [HHT$^+$16] we fitted each face to the Surrey's Face Model and extracted its respective 63 3DMM parameters. The expression parameters are ignored since we do not care about the person's active expression, since it does not contribute to age estimation and will only make the gathered information noisier.

4.3.3 3D Face Reconstruction and U.V. Maps

A facial 3D point cloud contains information about multiple details of a person's face. It can also present a level of detail not possible in a picture, such as detailing the sharpeness of the jaw and nose and depthness of the eye sockets. Along with a texture, the 3D facial mesh is the best digital representation of a human's face.

According to [FWS$^+$18], a U.V. position map is "a 2D image recording 3D positions of all points in U.V .space". Over the past few years, U.V. Mapping has been utilized to represent information in a 3D space such as texture of faces, 2.5D geometry and 3D facial mesh correspondence. [FWS$^+$18] uses U.V. space to store 3D coordinates of points of a 3D face model by defining a 3D point cloud in a Cartesian 2D coordinate system. The ground truth 3D face point cloud exactly matches the face in the 2D image when projected to the x-y plane. More details about U.V. Map representation is available in Section 3.

Lately, 3D reconstruction has been trending in the direction of dropping the 3DMM fitting. Instead, the point cloud is regressed directly from the 2D image. By bypassing the 3DMM fitting, the regressed face models are not only faster to estimate since there is no need to render the model and then refine it, the models are also not constrained by the original 3DMM model's parametrization.

For this reason, using [FWS$^+$18]'s method by adapting their publicly made available code into our pipeline on Github, we recover both a U.V. Map and a point cloud representation of every image in our datasets. The U.V. Map is a 2D square image of size 256 by 256 and 3 channels while the point cloud contains 43,868 points and its respective RGB encoded colour.

4.3.4 Orientation Renderings

Having the 3D point cloud of the face depicted in the picture means we have a 3D representation of the face of the individual present in the image. Originally, only one 2D projection of the face was available which was the input 2D image of the original dataset. However, since now we possess a synthetically created 3D version of the face, we can create new synthetic 2D projections of the face in orientations previously not available. Therefore, by rendering the 3D point cloud in different perspectives we can create multiple 2D images of the same person but by rendering the same point cloud in multiple perspectives we can get a system more robust to head poses and hopefully more efficient in the age estimation task. It has been shown that training deep learning models with synthetic data, even in different domains such as 3D landmark estimation [BT17b] or in-car object detection [TPA$^+$18] where new synthetic cars were created using Unreal Engine 4 and synthetically added them into pre-existing scenarios, can help improve performance and avoid the collection of large amounts of data.

In order to do this, we first need to centre the point cloud on the origin and make sure the person has his/her eyes and nose facing the camera, which we will call being frontalized. The point clouds provided by [FWS$^+$18] are not necessarily frontalized since it captures the head pose. Since all vertices are in full correspondence, for each point cloud we use a default frontalized point cloud centred in the origin and by using the least-squares solution of a linear matrix equation, we find a vector that solves 4.1, where A is the original point cloud and B is the default point cloud. Then apply the optimal affine transformation by subtracting the vector x of the original point cloud to get the centred and frontalized version of the original point cloud.

$$\underset{x}{argmin}\ ||B-Ax||^2 \tag{4.1}$$

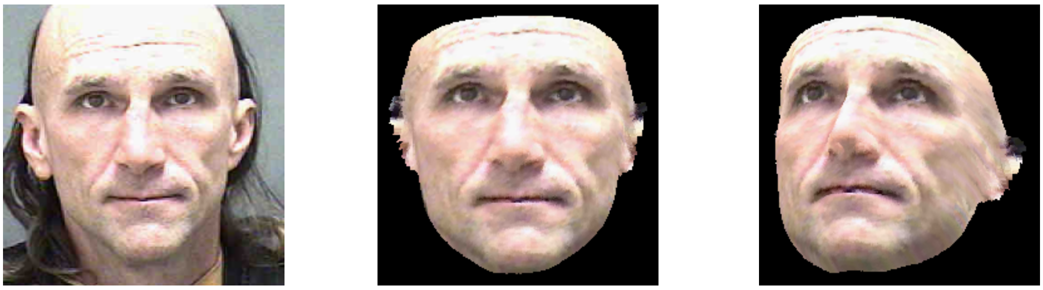
After having every point cloud frontalized, we apply a combination of rotations on all 3 axis. We apply rotations 4.2 4.3 4.4 to the point cloud in the range of -15^{o} to 15^{o} with a step of 15^{o} in every axis.

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\theta & -sin(\theta) \\ 0 & sin(\theta) & cos(\theta) \end{bmatrix} \tag{4.2}$$

$$R_y(\theta) = \begin{bmatrix} cos(\theta) & 0 & sin(\theta) \\ 0 & 1 & 0 \\ -sin(\theta) & 0 & cos(\theta) \end{bmatrix} \tag{4.3}$$

$$R_z(\theta) = \begin{bmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.4}$$

These transformations are iteratively applied to a centred point cloud (now it is not necessary for the point cloud to be frontalized). Therefore, after every transformation the point cloud is centred by subtracting to each coordinate the mid point between the maximum and minimum value of the meshes coordinate points of the corresponding axis. Every mesh will be rotated 27 different ways and then rendered into 2D using the z-buffer algorithm, with a black background. This means we augmented our original image by a factor of 27, when we count the model rendered without any rotation. Examples of this process can be seen in Figure 4.10.



(a) Original image (b) No Transformation. (c) Rotated Projection.

Figure 4.10: Example of the transformation process.

4.3.5 Summary of Pre Processing Pipeline

The dataset pre processing pipeline is summarized in Figure 4.11. Every green block indicates information to be used by a CNN model.

4.4 Problem Formulation

According to recent studies [RTG15, HHM+17, ZLY+18], although age estimation is a regression problem since age is sampled from a continuous range of values, it has been found that formulating it as a classification problem results in a better overall performance. This might happen due to only sparse age values being available in datasets. In other words, although age is a continuous values, only rounded age values are available, therefore making them discrete values in practice. [RTG15] proposed a classification formulation by defining age classes from 0 to 100. This 101 classes are picked from preliminary experiments and it is claimed to be the best choice out of those experiments.

A softmax expected value (3.5) is performed on the output in order to obtain more precise results and improve the overall accuracy of the predictions.

4.5 Models

4.5.1 Baseline

In order to effectively study how the addition of 3D features affect the performance of an age estimation model, we first create a baseline model which will be trained on 2D images only. This way, by comparing our results to this model's performance, we can measure the improvement resulting from the addition of the 3D features. It is also important to point out that the baseline model is much smaller than recent state of the art approaches and the same is true for consequent mutations of this model. Building a state of the art approach is out of the scope of this thesis, instead we seek to prove that adding 3D based facial features can improve currently existing approaches. The choice for the lack of complexity of the model exists mainly due to lack of computational resources. By having a simpler model, it is possible to test more hypothesis and cover a wider range of 3D feature representations and their effect on the age estimation task.

Our baseline model will follow [RTG15] model architecture, which is based on the VGG16 network architecture [SZ14]. The network has sixteen layers, thirteen of which are convolutional, hence the name. In this case, after each convolution we apply Zero Padding to our feature map, in order to maintain spatial dimensions. As has been stated before, we would like to have a simpler model and for that reason we remove the all fully connected layers but the last. We also follow the approach of [RTG15] by changing the output of our model to 101 classes. This transformations reduce the original 138,357,544 million parameters of a standard VGG16 network to 17,248,677. An overview of the baseline model can be seen in Figure 4.12.

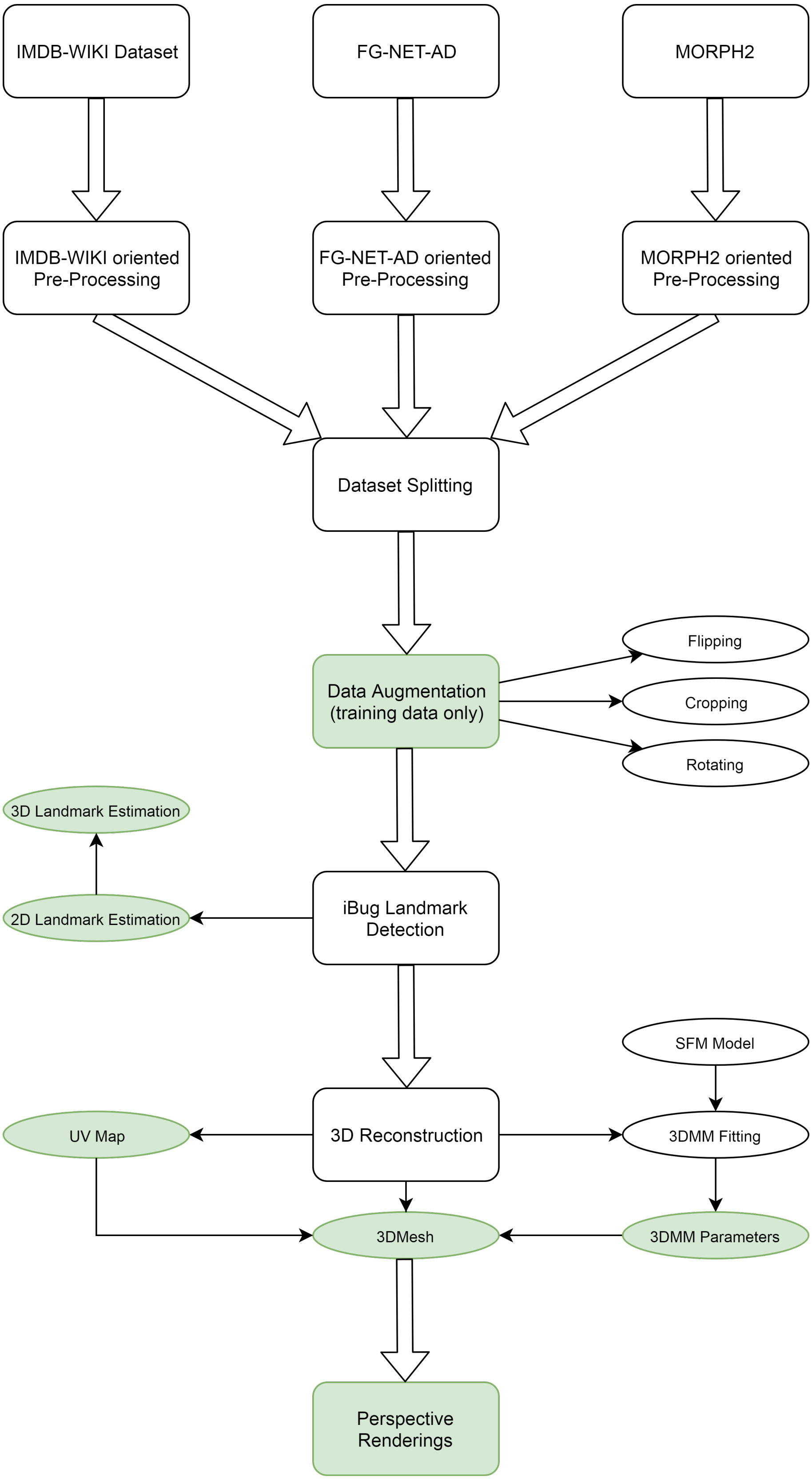


Figure 4.11: Summary of Dataset Pipeline.

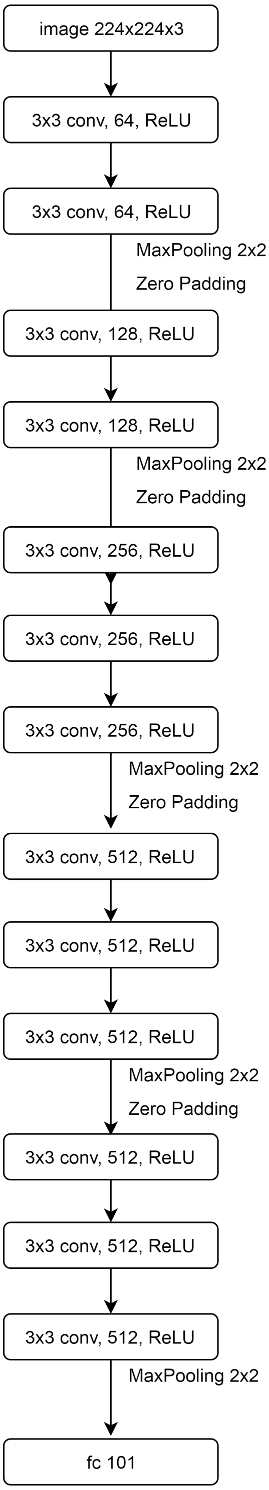


Figure 4.12: The architecture of the Baseline Model. Zero Padding is applied before each Convolutional Layer.

Every model iteration will be based on this model and although we will extend this model by adding feature extraction layers from other types of data, we will not change the fully connected layers. This is to make sure we don't make our model capable of processing more complex tasks. We want to compare how 3D features can improve the current state of age estimation, not create a perfected model to work with this data. By not changing the fully connected layers we also avoid having big shifts in the amount of parameters of the models.

Since our model follows the same CNN architecture as [RTG15], their task is similar to ours and they made their trained models available on their website, we use their CNN pretrained weights as our initializer. However, the same is not true for the fully connected layers, as the extracted features differ from their work. We initialize our own layers using He's distribution [HZRS15b].

4.5.2 Landmark Models

Firstly, we create a model able to receive as input both 2D face images and respective landmarks, whether these landmarks contain 2D or 3D information. In order to do this, after the image feature extraction, we concatenate our landmark data to those features, which can add model stability. The landmark information of each image is standardized by dimension. The fully connected layers expect a 1D feature vector therefore our landmarks are squashed to a one dimensional vector. This results in our model containing an extra 13,736 or 20,604, in case of 2D or 3D landmark information respectively, extra features which translates to an overall number of parameters of 17,262,413 and 17,269,281, for 2D and 3D landmarks models respectively. An overview of the Landmark model is available in the Appendix and can be seen in Figure A.1.

4.5.3 U.V. Maps

We use this 3D facial representation as an additional input of our baseline model. Parallel to the face CNN feature extraction of our baseline model, we run a small CNN through the U.V. Maps of the corresponding face. We then concatenate the 3D extracted features to the output of our original baseline CNN in the same fashion as the previous models.

The new concatenated CNN is going to be composed by eight Convolutional Layers, with a Max Pooling operation every two layers. The outline of this secondary CNN can be seen in Figure A.2. This network has an additional 2,155,152 over the baseline model, totalling 19,403,829.

4.5.4 3DMM Model

Analogous to our Landmark Model in Section 4.5.2, we concatenate the extracted 3DMM parameters with the CNN extracted features. Using the Surrey 3DMM parameters [HHT^{+}16], the resulting model will have an additional 6,363 parameters, resulting in an overall 17,255,040 number of parameters. An overview of the 3DMM model is available in the Appendix and can be seen in Figure A.3.

4.5.5 Cloud Point Model

As the previous Landmark Model and 3DMM Model, in section 4.5.2 and 4.5.4 respectively, we plan to concatenate the 3D points from the extracted meshes. However, the extracted meshes contain 43,868 3D points and respective RGB values. Introducing this ammount of points to our model would result in an increase of parameters by a factor of 2. Since we wish for our models to be similar and we want to reduce their complexity as much as possible, we subsample the extracted mesh. Since all points are in full correspondence and the mesh is extremely dense, we can safely sample every four points which results in a subset of 10,967 3D points. Each cloud point is then standardized by dimension, similarly to the landmarks. The resulting model will have an additional 20,571,678 parameters. An overview of the Cloud Point model is available in the Appendix and can be seen in Figure A.4.

4.5.6 Renderings Model

Using the same architecture as the baseline model, this model will differ from the baseline purely on the input domain. In contrast to the 2D images real images(although augmented) received by the baseline model, this model receives the rendered images described in Section 4.3.4. By being identical to our baseline model, it contains the same number of parameters.

However, unlike with the previous models, we can evaluate the result from our renderings in a different, more robust manner. Since we possess 27 renderings of the same person in the same pose, instead of treating each image differently and in practice use these images as a data augmentation technique, we can use all of them and get the average estimated age, which in turn is obtained using the softmax expected value 3.5.

We can take this idea even further. We expect to have some perspectives introducing larger errors than others. With this in mind, we purpose a weighted average of the rendering's estimation. The way we do this in order to not overfit to our training set is to use the validation set. So we run the model for every rendering in the validation set and sum the errors that occur in each perspective for every image. We then apply a softmax normalization operation in order to obtain the weight that each perspective has in the overall error.

However, in our weighted average we want perspectives with a higher error weight to have a lower impact. Therefore we apply a symmetry operation and sum 1. The whole process can be seen in Equation (4.5):

$$\begin{aligned} E(\hat{y},y)^{(p)} &= \sum_{i=0}^{N} |\hat{y}_i^{(p)} - y_i^{(p)}| \\ W(\hat{y},y) &= 1 - \sigma(E(\hat{y},y)) \end{aligned} \tag{4.5}$$

where p is a distinct perspective, $E(\hat{y},y)^{(p)}$ is the accumulated MAE over N images of perspective p and σ is the softmax operation (2.13).

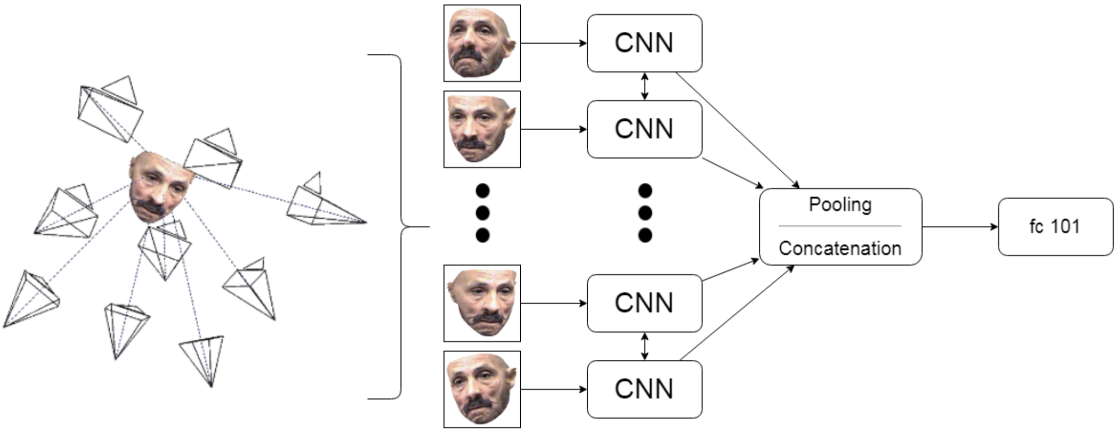


Figure 4.13: The architecture of the Multi-View Rendering Model, inspired by [SMKL15].

4.5.6.1 Multi-View Rendering Model

Extending the idea of using different perspectives to estimate the age of a person, we implement a multi-view CNN model that takes as input more than one perspective at a time. This allows the model to have global information about the 3D face model and use this 3D knowledge to infer that persons' age. This model is inspired by the Multi-View CNN for 3D Shape Recognition [SMKL15]. The idea behind this model is to take the estimated 3D shape descriptors from each view and combine them in to create a stronger descriptor. [SMKL15] suggests an element-wise Maxpooling or Avgpooling operation across all views but pointing out that their empirical observations support a preference for Maxpooling. Due to this pooling operation, the multiview image descriptor will have the same size as the baseline model's image descriptor which means both models have the same number of parameters.

Although this idea is applied for voxel grids and polygon meshes, the nature of a multiview system suggest that the strengths of this model can be applied on other multiview based tasks such as this. Our implementation relies on extracting a facial feature vector from each view, combine those features to create a robust representation of the facial features and then use those features to extract the age value.

Moreover, we feel like the implementation of pooling operation might inadvertently destroy key features present it the image. For this reason and in order to remain similar to other models in this project, instead of a pooling operation we perform a concatenation operation which allows us to maintain the extracted features of each view when performing a global operation on the fully connected layer.

An overview of the Multi-View Rendering Model can be seen in Figure 4.13.

4.6 Implementation

There are several exiting deep learning frameworks to help speed up development and research. These tools provide commonly used components such as automatic differentiation, integrated GPU computation and a community of users that support, maintain and contribute to the continuous improvement of each tool. Among these tools are:

- Tensorflow [AAB+15]
- PyTorch [PGC+17]
- Caffe and Caffe2 [Jia13]
- Scikit-learn [PVG+11]
- MXNet [CLL+15]
- CNTK [SA16]

We chose to implement our project using the high-level Deep Learning library Keras [C+15] for Python 3, due to its rapid prototyping capabilities, using the Tensorflow [AAB+15] backend, which is often directly used to create custom layers and loss functions.

To extract the Caffe [Jia13] model weights from the DEX [RTG15] pre-trained model and convert them into Keras [C+15] we used the tool *caffe2keras* available in Github. We also use [FWS+18]'s 3D reconstruction publicly available implementation in order to extract point cloud information and respective rendering and the EOS [HHT+16] tool to extract 3DMM parameters.

The OpenCV Library [Bra00] was used to perform mainly image pre-processing, Matplotlib with Seaborn for graphical visualisation purposes and Numpy for scientific computation.

4.6.1 Training Configuration

Each model is trained with similar parameters:

- **Number of Epochs**: In machine learning, one epoch means a full cycle of training where every sample of the training set was used. We will train each model for 20 epochs.
- **Normalization**: L2 Weight Normalization with $\lambda = 0.0005$ weight on the loss function.
- **Optimization Algorithm**: We follow [RTG15], and others [HHM+17, ZLY+18], and use Mini-Batch Gradient Descent, with batches of 32 instance of data, except for the Multi-View which is trained with SGD. The learning rate was set as 0.0001 with Nesterov momentum [SMDH13], where $\alpha = 0.9$ was set, hyper-parameters also used by [RTG15, HHM+17].
- **Model Checkpoints**: At the end of each epoch we will validate it using our validation set. We will save the current state of the model's parameters as well as it's performance on the validation data.
- **Weight Freezing**: What is commonly called "Weight Freezing" refers to action of preventing an update of specific weights during the back propagation phase, making them constant

or frozen. [YCBL14] suggests to not freeze any layer as it doesn't seem to bring any advantages other speeding up the training process. By not freezing the layers, we allow layers to learn new co-dependences instead of enforcing the exiting ones due to pre-training. For this reason we do not freeze any layer.

The models were trained on Unix based Workstation with an Intel Core i9-7900X CPU running at 3.30 GHz, with 64 Gb of RAM and a GeForce GTX 1080 TI NVIDIA Graphics Card. I would like to thank Jaguar Land Rover, specially the Research Team, for allowing me to use their machines during downtime.

Chapter 5

Results

In this chapter, we will present the results of each model.In order to evaluate each model's performance we will use the MAE (2.15) in years. This is the standard metric when evaluating age estimation system [RTG15, HHM^{+}17, ZLY^{+}18, ABBD16]. By comparing the MAE results we can determine which model's perform worse on average compared to others where a lower MAE translates to an overall better model.

For the Renderings Model of Section 4.5.6, we will present 3 different metrics: the standard per image prediction MAE, the MAE of the average prediction of every perspective's rendering for a single person and the MAE of the weighted average of every perspective's rendering, taking into account the average error from each perspective, for a single person, described by Equation (4.5).

5.1 Model Results

5.1.1 IMDB-WIKI

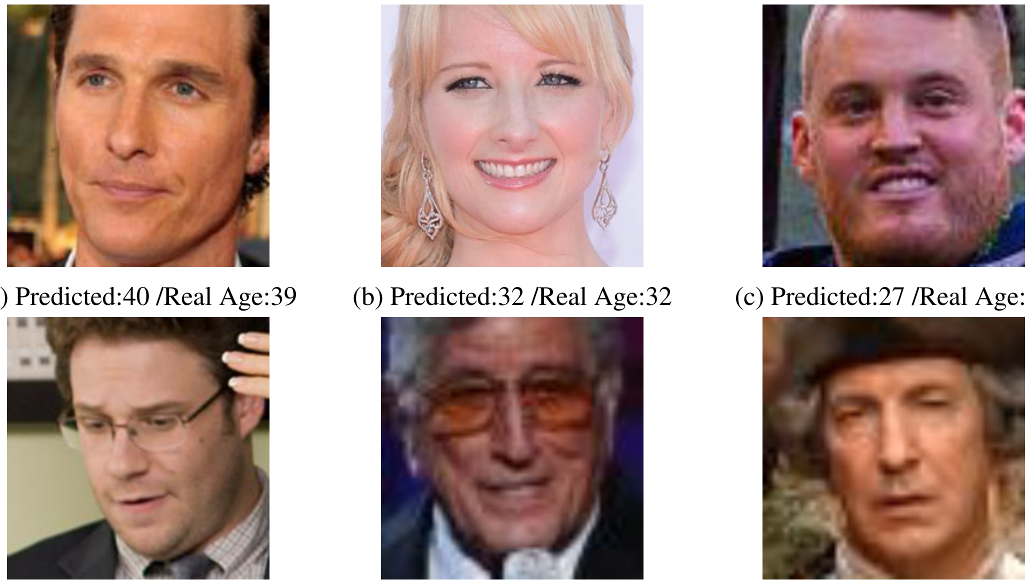
Table 5.1 details the results of training from scratch with the IMDB-WIKI dataset. Each model's performance is individually depicted using the MAE metric.

Table 5.1: MAE comparison of the models on IMDB-WIKI test set.

Model Type	**IMDB-WIKI (Test Set)**
Baseline	5.30
2D Landmarks	5.24
3D Landmarks	5.13
U.V. Maps	5.28
Point Cloud	5.19
3DMM	5.23
Renderings	5.71/5.40/5.39
Multi-View (Pooling)	6.12
Multi-View (Concatenation)	6.19

The results are as predicted and models supporting additional 3D inputs outperform the baseline. This indicates a important aspect of the synthetic data. The additional 3D information extracted from each image has significant value to contribute to the better estimation of age. This seems to indicate that the shape of the face does contain pertinent information regarding the person's age. Note that the additional information does not guide the originally available information on the 2D image.

However, we fail to see any improvement when using the Renderings and Multi-View models. This might occur due to a number of factors but it is most likely due to the low quality image availability. As mentioned in Section 4.1.1, this dataset is very noisy even after severe quality aimed sub-sampling. This lack of quality affects the subsequent data rendering process, resulting in low quality renders.



(a) Predicted:40 /Real Age:39

(b) Predicted:32 /Real Age:32

(c) Predicted:27 /Real Age:27

(d) Predicted:30 /Real Age:70

(e) Predicted:51 /Real Age:11

(f) Predicted:65 /Real Age:15

Figure 5.1: Examples of the best (first row) and worst (second row) prediction results of the Baseline model on the IMDB-WIKI dataset.

In Figure 5.1 it is possible to see examples of 2D images where the baseline model had more difficulties or ease when estimating a person's age. It can be seen that the neither the quality of the image or the its label are very good resulting in a negative impact on our models ability to learn.

5.1.2 IMDB-WIKI Pre-training Results

As mentioned in Section 4.5.1, since we change our fully connected section differs from the one used in DEX [RTG15], we were unable to use their available pre-trained weights. Using pre-trained weights help models converge faster and better generalize to new data [YCBL14].

However, since we train each model from scratch using the IMDB-WIKI dataset, we can compare the use of our own pre-trained weights with the training from scratch. So evaluating our models on MORPH2 and FG-NET-AD, we train each model twice. Once with the last layer trained from scratch and the Convolutional Layers' weights initialized with [RTG15]'s weights and then a second time using our the weights from our models trained with the IMDB-WIKI dataset.

The columns on Tables 5.2 and 5.3 with an indicated (P) refer to the models fully pre-trained and these models are trained as described in Section 4.6.

5.1.3 FG-NET-AD

Table 5.2 details the results of training from scratch with the FG-NET-AD dataset. Each model's performance is individually depicted using the MAE metric.

Table 5.2: MAE comparison of the models on FG-NET-AD test set.

Model Type	FG-NET-AD	FG-NET-AD (P)
Baseline	3.86	3.15
2D Landmarks	3.85	2.92
3D Landmarks	4.03	3.00
U.V. Maps	4.09	3.09
Point Cloud	4.29	3.03
3DMM	4.12	3.07
Renderings	3.86/3.69/3.67	3.31/3.02/3.02
Multi-View (Pooling)	4.57	3.66
Multi-View (Concatenation)	5.03	4.23
DEX [RTG15]	4.63*	3.25*

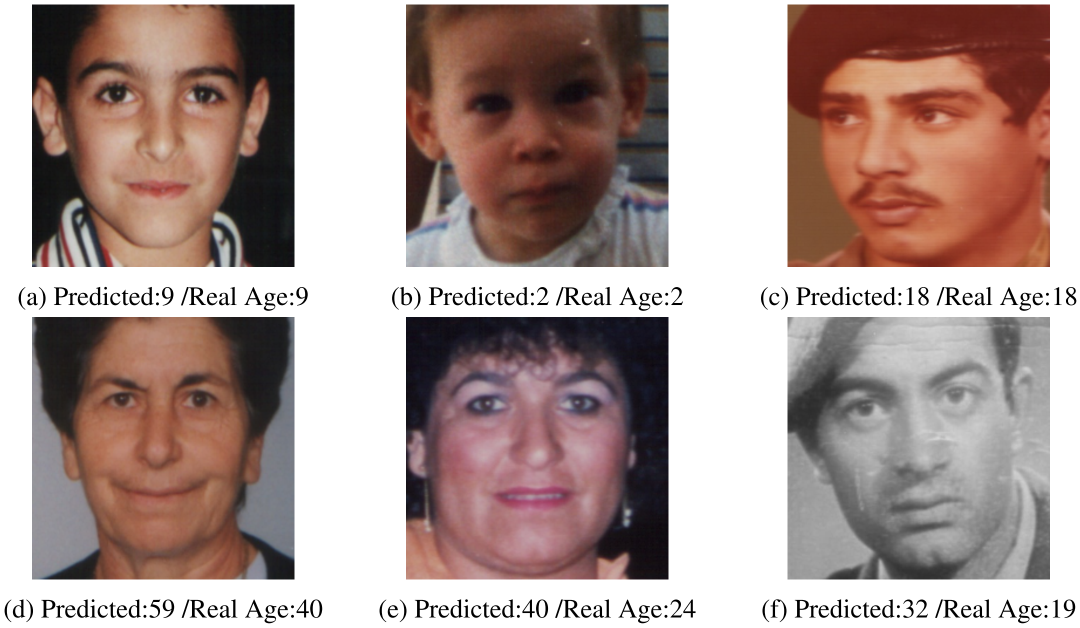
We mark [RTG15] with an asterisk to note the usage of a different validation technique, we use 20% of the available dataset for testing purposes while [RTG15] perform Leave One Out Cross-Validation.

The first thing to notice when analysing Table 5.2 is the improvement brought on by the usage of pre-trained weights. This was to be expected since the study done by [RTG15] proves this to be the case.

Further analysing Table 5.2, shows us that most of the improvements that occurred when evaluating the models on the IMDB-WIKI dataset translated themselves onto the FG-NET-AD. However, this only holds up when looking at the models with Pre-Trained weights. When looking at the models trained from scratch, we can conclude that they were not capable of interpreting 3D information. The most plausible reason for this is lack of variety in the training data, even after augmentation techniques, and we use pre-trained weights to address this issue.

Unlike previously, the model struggle to learn to interpret 3D landmarks as well as the previous dataset. Although we cannot know for certain without further study, grayscale colour aspects may be the cause for this. 2D Landmark estimation is a much more mature field than 3D Landmark estimation which has its negative impact on the quality of the extract synthetic data and consequently in the performance of the model.

However, we find the predictions of the Rendering Model to actually be better than the the baseline ones, by a significant margin.



(a) Predicted:9 /Real Age:9

(b) Predicted:2 /Real Age:2

(c) Predicted:18 /Real Age:18

(d) Predicted:59 /Real Age:40

(e) Predicted:40 /Real Age:24

(f) Predicted:32 /Real Age:19

Figure 5.2: Examples of the best (first row) and worst (second row) prediction results of the Baseline model on the FG-NET-AD dataset.

In Figure 5.2, we can see where the Baseline model shines best and where it fails. It is interesting to see a grayscale image as one of the more difficult images in the test set. Grayscale images are not abundant in the dataset so it may be interpreted as noise and makes the age classification more difficult. As stated in Section 4.1.3, we are more likely to see grayscale images the further down the age range we go. Therefore, Figure 5.2f shows an indication of how the model might have overfitted grayscale images as young ages, demonstrated by the wrongful predictions of a much younger age.

5.1.4 MORPH2

Table 5.3 details the results of training from scratch with the FGNET dataset. Each model's performance is individually depicted using the MAE metric.

As expected, we see a clear boost in performance when using IMDB-WIKI pretrained weights. However, unlike the experiments, we fail to see any improvement over the baseline when adding 3D information. As described in Section 4.1.1, MORPH2 is the dataset with the lowest image resolution of the three we tested. We believe that the data quality affects negatively the prediction and the pipeline tool's ability to generate realistic data. This is believed to be the reason behind the contrasting results.

In Figure 5.3 we can see examples of the most difficult predictions of the MORPH2 Baseline model, which are extremely complicated cases even for a human to deal with.

Table 5.3: MAE comparison of the models on MORPH2 test set.

Model Type	MORPH2	MORPH2 (P)
Baseline	2.79	2.73
2D Landmarks	2.83	2.73
3D Landmarks	2.86	2.82
U.V. Maps	4.09	3.09
Point Cloud	2.86	2.78
3DMM	2.84	2.76
Renderings	3.08/2.90/2.89	2.99/2.81/2.80
Multi-View (Pooling)	3.28	3.20
Multi-View (Concatenation)	3.42	3.37
DEX [RTG15]	3.09	2.68

Our Baseline model performs slightly worse than the model which is inspired its creation [RTG15] while having 13% of its ancestor's parameter count. This signals an overcomplexity of the original model and is the reason why it is more prone to overfitting.

5.2 Cross Dataset Results

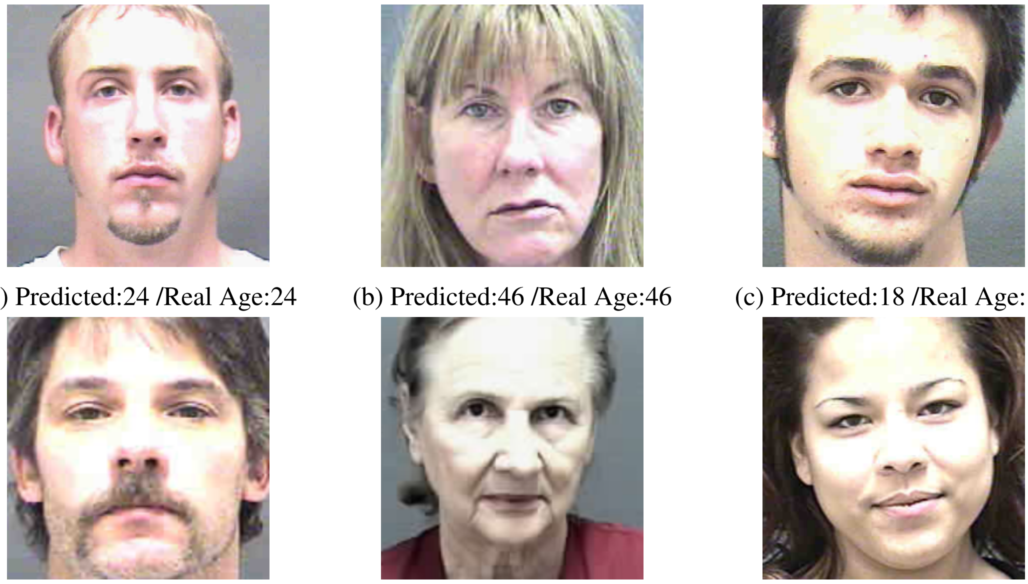
In this section, we will explore the use of models which were trained on sampled images from different distributions. The objective is too show how well models generalize to other datasets. Table 5.4 shows the results from the cross examination for the baseline models. Models marked with *(P)* are the aforementioned models trained on IMDB-WIKI pre-train weights.

Table 5.4: Cross Dataset Results for the Baseline model.

Dataset	IMDB-WIKI	FG-NET-AD (P)	FG-NET-AD	MORPH2 (P)	MORPH2
IMDBWIKI	-	7.71	9.94	6.97	7.58
FG-NET-AD	9.14	-	-	11.9	13.45
MORPH2	4.40	4.98	10.82	-	-

From Table 5.4 we can observe that models trained with datasets with similar distributions are more robust to cross dataset examination. Models trained with MORPH2 or IMDB-WIKI are more easily interchangeable than any of those datasets with FG-NET-AD. This can also be a sign of overfitting to the age distribution, issue that tends to happen frequently in the age estimation task and explains why FG-NET-AD models behave poorly and other models behave poorly on FG-NET-AD, as the FG-NET-AD peak in the distribution, which can be seen in Figure 4.2, is of 10 year olds, an age that is outside MORPH2's age range.

The impact of the data variety is also noticeable, as it possible to see that every model that was trained under pre-trained weights has a better performance than the respective trained from scratch counter-parts. This fact defends the aforementioned decision to use IMDB-WIKI pre-trained weights.



(a) Predicted:24 /Real Age:24

(b) Predicted:46 /Real Age:46

(c) Predicted:18 /Real Age:18

(d) Predicted:52 /Real Age:36

(e) Predicted:57 /Real Age:74

(f) Predicted:34 /Real Age:18

Figure 5.3: Examples of the best (first row) and worst (second row) prediction results of the Baseline model on the MORPH2 dataset.

Table 5.5: Cross Dataset Results for the Renderings model (Weighted Average Only).

Dataset	**IMDB-WIKI**	**FG-NET-AD (P)**	**FG-NET-AD**	**MORPH2 (P)**	**MORPH2**
IMDBWIKI	-	10.55	8.93	7.01	7.93
FG-NET-AD	7.75	-	-	11.50	13.46
MORPH2	4.75	5.80	8.53	-	-

Cross Dataset results from Table 5.5 corroborates the previous statements concerning overfitting as we see again that other datasets fail to generalize to FG-NET-AD. Following the results from Sections 5.1.1, 5.1.3 and 5.1.4, we expected the Cross Dataset examination to show this model to have an overall worse performance than the baseline which is what we can observe.

However, when paying close attention to the results on the FG-NET-AD dataset, you can see that the MAE actually decreases from Table 5.4 which is another sign of the superior image quality of the FG-NET-AD dataset which allows us to create more realistic 3D face models and respective renderings. This shows that even if models are trained with lower quality renderings, they can still outperform traditional single image based models if fed with good enough data.

5.3 Age Grouping Results

The task of precisely estimating the real age of a person is important in certain applications. However, other applications may not need this type of precision and instead be satisfied with an estimation with a larger range with a more robust accuracy. Some age estimations datasets were specifically designed for this grouped range age estimation, such as [EEH14] which use pre-defined ranges.

This project's implementation can determine the accuracy of our models using age ranges. It should be pointed out that although we analyse our results with age groups, our models are not altered in any way and still output a continuous value. This value is then bagged into a certain age range for the results to be displayed.

This project's implementation allows for manual or automatic distribution of ages into bags. We will present the results with 2 different ways of age grouping:

- **Pre-Defined Bagging**: We follow an age grouping that splits ages into groups of 10, with 10 bags overall. This bagging technique is independent of the dataset characteristics. However, some datasets have their median age shifted from 50 which may result in some bags having zero instances.

- **Dynamic Bagging**: Instead of pre-defining the bagging technique, we only manually set the number of age bags and then based on the dataset we are testing, we determine the age range of each bag based on the overall age range available in that dataset.

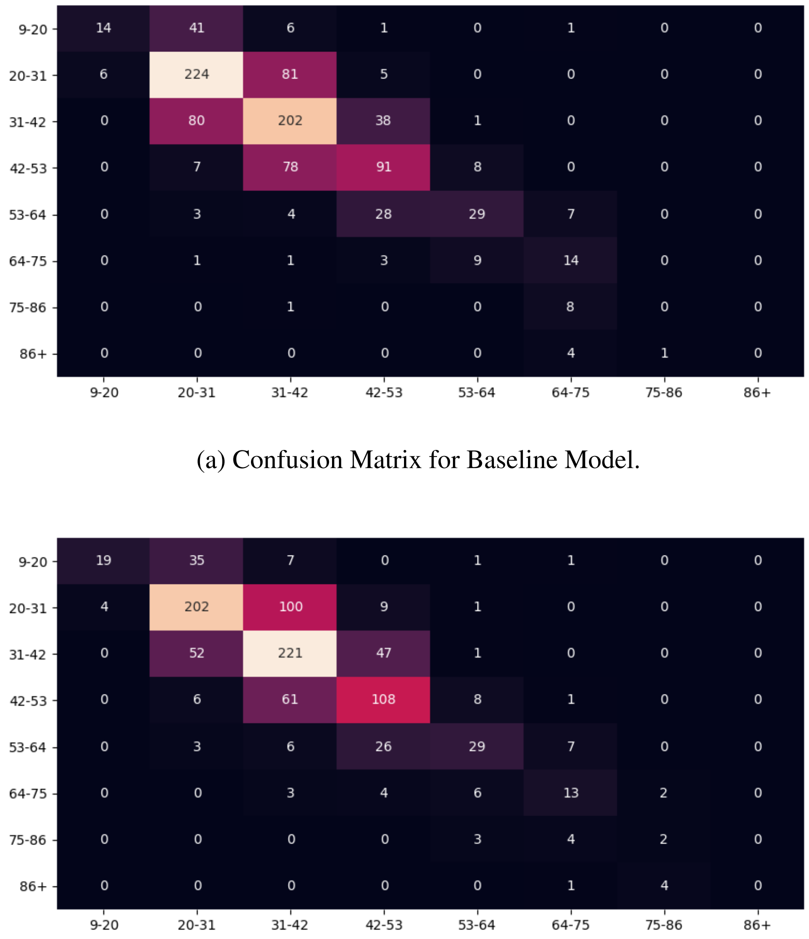
For each dataset we will only present the results of our Baseline and Rendering models using Dynamic Bagging with 8 bags. Due to space limitations, we will only display the dynamic bagging results in this section. The Renderings model results will be based on its weighted average metric. The Pre-Defined Bagging results in the Appendix, on section B.

We will also present the results for accuracy per bag, 1-off accuracy per bag and overall bagging accuracy. k-off accuracy accepts a wrong prediction as correct if it is contained on the *k* adjacent bags.

5.3.1 IMDB-WIKI

The FG-NET-AD confusion matrices for the Baseline and Rendering models are displayed in Figure 5.4. A Table 5.6 detailing the accuracy of the model given each age range is also available.

With dynamic bagging we achieve a 58% and 60% accuracy for the Baseline and Renderings model, respectively. It is interesting to see that although the Renderings model has a higher MAE, it actually results in a higher accuracy. This suggests that the Renderings model might be more precise than the Baseline. The Baseline's 1-off accuracy is however higher at 96% while the Renderings model achieves 95%.



	9-20	20-31	31-42	42-53	53-64	64-75	75-86	86+
9-20	14	41	6	1	0	1	0	0
20-31	6	224	81	5	0	0	0	0
31-42	0	80	202	38	1	0	0	0
42-53	0	7	78	91	8	0	0	0
53-64	0	3	4	28	29	7	0	0
64-75	0	1	1	3	9	14	0	0
75-86	0	0	1	0	0	8	0	0
86+	0	0	0	0	0	4	1	0

(a) Confusion Matrix for Baseline Model.

	9-20	20-31	31-42	42-53	53-64	64-75	75-86	86+
9-20	19	35	7	0	1	1	0	0
20-31	4	202	100	9	1	0	0	0
31-42	0	52	221	47	1	0	0	0
42-53	0	6	61	108	8	1	0	0
53-64	0	3	6	26	29	7	0	0
64-75	0	0	3	4	6	13	2	0
75-86	0	0	0	0	3	4	2	0
86+	0	0	0	0	0	1	4	0

(b) Confusion Matrix for Renderings Model.

Figure 5.4: Confusion Matrices on IMDB-WIKI dataset.

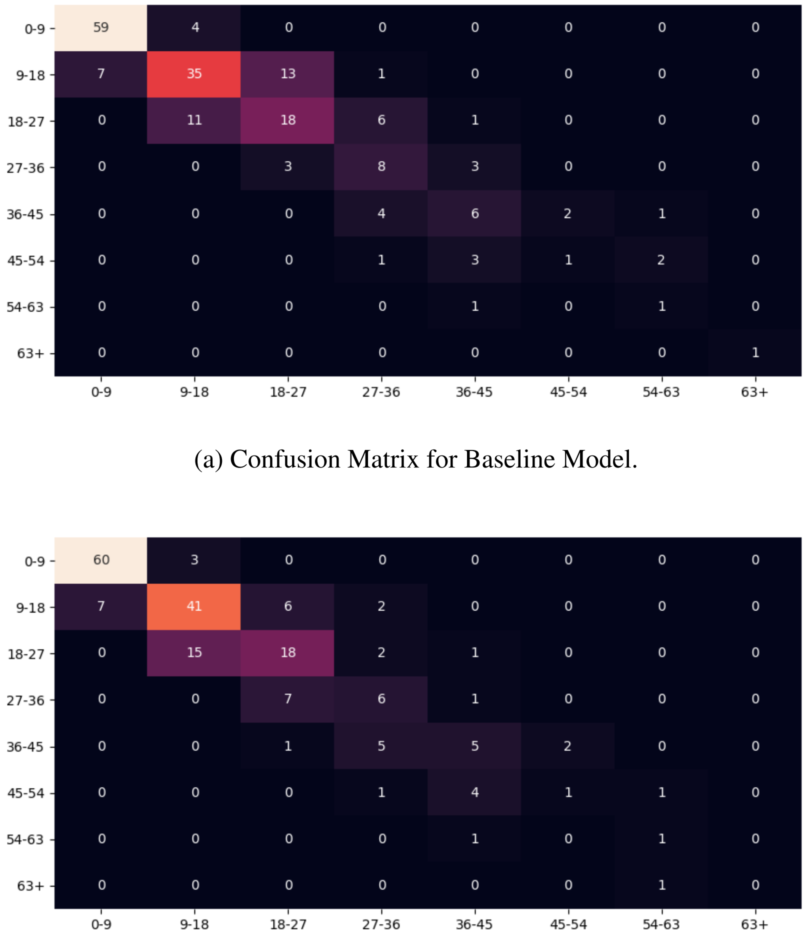
Table 5.6: Accuracy of each model for each age range on the IMDB-WIKI dataset.

Type of Accuracy	**[0-9[**	**[9-18[**	**[18-27[**	**[27-36[**	**[36-45[**	**[45-54[**	**[54-63[**	**[63-[**
Baseline - Exact Accuracy	0.222	0.709	0.629	0.495	0.408	0.5	0.0	0.0
Baseline - 1-Off Accuracy	0.873	0.984	0.997	0.962	0.901	0.821	0.889	0.2
Renderings - Exact Accuracy	0.302	0.639	0.688	0.59	0.408	0.464	0.222	0.0
Renderings - 1-Off Accuracy	0.857	0.968	0.997	0.962	0.873	0.75	0.667	0.8

Table 5.6 and the Confusion Matrix in Figure 5.4 show that both models struggles the most with age ranges at both extremes of the spectre. However, we can observe that the Renderings model is more robust to older ages while the Baseline is more robust to younger ones.

5.3.2 FG-NET-AD

The FG-NET-AD confusion matrices for the Baseline and Rendering models are displayed in Figure 5.5. A Table 5.7 detailing the accuracy of the model given each age range is also available.



	0-9	9-18	18-27	27-36	36-45	45-54	54-63	63+
0-9	59	4	0	0	0	0	0	0
9-18	7	35	13	1	0	0	0	0
18-27	0	11	18	6	1	0	0	0
27-36	0	0	3	8	3	0	0	0
36-45	0	0	0	4	6	2	1	0
45-54	0	0	0	1	3	1	2	0
54-63	0	0	0	0	1	0	1	0
63+	0	0	0	0	0	0	0	1

(a) Confusion Matrix for Baseline Model.

	0-9	9-18	18-27	27-36	36-45	45-54	54-63	63+
0-9	60	3	0	0	0	0	0	0
9-18	7	41	6	2	0	0	0	0
18-27	0	15	18	2	1	0	0	0
27-36	0	0	7	6	1	0	0	0
36-45	0	0	1	5	5	2	0	0
45-54	0	0	0	1	4	1	1	0
54-63	0	0	0	0	1	0	1	0
63+	0	0	0	0	0	0	1	0

(b) Confusion Matrix for Renderings Model.

Figure 5.5: Confusion Matrices on FG-NET-AD dataset.

By analysing both Table 5.7 and the Figure 5.5 that the age range the model struggles the most is with people with ages between 45-63. This struggle is mostly due to poor dataset balance. Lack of examples of people with ages in the range of 45-63 is reflected on the model's poor performance.

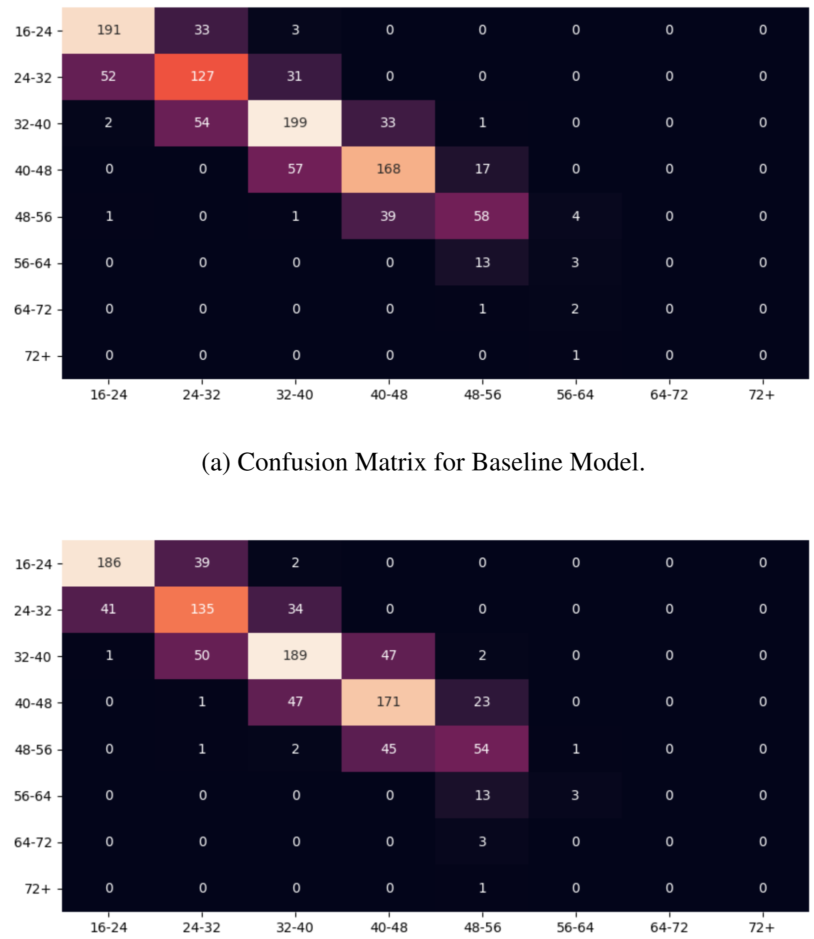
We can also report an overall accuracy of 67% and 69% for the Baseline and Renderings model, respectively. We expect this to be the case since we the MAE for the Renderings model is inferior to Baseline error and it corroborates our findings on the IMDB-WIKI dataset. Moreover, we report another 1% increase in favour of the Baseline model when comparing the 1-Off Accuracy with the Renderings Model, which goes in line with what was previously found, with the Baseline Model having 97% 1-Off Accuracy while the Renderings model has 96%.

Table 5.7: Accuracy of each model for each age range on the FG-NET-AD dataset.

Type of Accuracy	[0-9[	[9-18[	[18-27[	[27-36[	[36-45[	[45-54[	[54-63[	[63-[
Baseline - Exact Accuracy	0.937	0.63	0.5	0.572	0.462	0.143	0.5	1.0
Baseline - 1-Off Accuracy	1.0	0.982	0.972	1.0	0.923	0.857	0.5	1.0
Renderings - Exact Accuracy	0.952	0.732	0.5	0.429	0.385	0.143	0.5	0.0
Renderings - 1-Off Accuracy	1.0	0.965	0.972	1.0	0.923	0.857	0.5	1.0

5.3.3 MORPH2

The MORPH2 confusion matrices for the Baseline and Rendering models are displayed in Figure 5.6. A Table 5.8 detailing the accuracy of the model given each age range is also available.



(a) Confusion Matrix for Baseline Model.

(b) Confusion Matrix for Renderings Model.

Figure 5.6: Confusion Matrices on MORPH2 dataset.

Table 5.8: Accuracy of each model for each age range on the MORPH2 dataset.

Type of Accuracy	**[0-9[**	**[9-18[**	**[18-27[**	**[27-36[**	**[36-45[**	**[45-54[**	**[54-63[**	**[63-[**
Baseline - Exact Accuracy	0.841	0.605	0.689	0.694	0.563	0.188	0.0	0.0
Baseline - 1-Off Accuracy	0.987	1.0	0.99	1.0	0.981	1.0	0.667	0.0
Renderings - Exact Accuracy	0.819	0.643	0.654	0.707	0.524	0.186	0.0	0.0
Renderings - 1-Off Accuracy	0.991	1.0	0.99	0.996	0.971	1.0	0.0	0.0

The results on the MORPH2 dataset are as expected with the Renderings model performing slight worse than the Baseline. We previously stated that the reason behind this performance is the quality of the images provided by this dataset.

Unlike the previous datasets, both Overall Accuracy and 1-Off Accuracy are the same for both datasets, with 68% Overall Accuracy and 99% 1-Off Accuracy.

A brief analysis both the Table 5.8 and the Figure 5.6 leads to the similar conclusions stated in Sections 5.3.1 and 5.3.2 which tells us that the models struggle most with age ranges not prevalent in the datasets, which was previously noted during the cross-dataset analysis in Section 5.2.

Chapter 6

Conclusions

6.1 Main Contributions

The main contribution of this research work was to show the role of 3D facial information in the enhancement of current age estimation systems.

The first problem we tackled was the lack of significant data as there is no age focused dataset with 3D facial information publicly available. We overcome this problem successfully by creating a synthetic dataset based on current 2D image age datasets. Our pipeline allows for an effortlessly and personalized creation of the new instances of synthetic data, generated from state of the art publicly available tools which are seamlessly integrated in our work.

We started by collecting three different datasets, the most influential on the age estimation field. Each dataset was then pre-processed and augmented according to its own specificities. Afterwards, we extracted facial features such as landmarks, 3DMM parameters and 3D point cloud. The last one was subsequently used to create a new synthetic dataset composed of renderings of the point cloud from different camera positions. A tool was designed and implemented in order to perform the previously mentioned steps automatically.

We designed, implemented and evaluated several deep learning models aimed at extending the age estimation task with 3D facial features. Our contributions include a novel weighted averaging technique for multi-perspective estimation and a Multi-View model which combines information from multiple perspectives into a single 3D face descriptor.

We presented empirical results supporting the use of additional 3D information to enhance age estimating systems. Both the presented Renderings and Multi-View models are trained with fully synthetic data and still perform up to par with state of the art models trained with real data. We also showed the importance of data variety and pre-training methodology. Without using weights pre-trained on the IMDB-WIKI dataset, we would not have found good results when applying our methods to both FG-NET-AD and MORPH2 datasets.

However, we showed that these model's performance are conditioned on the quality of the data they are trained on, specially for perspective based models. We also present a cross-dataset evaluation which corroborates the model's quality dependence while highlighting overfitting challenges in the field of age estimation.

6.2 Future Work

Our data pipeline tool was created with extendibility in mind. The data pipeline can be extended to include extraction of different kind of data and different data augmentation techniques and data balancing techniques. It can also be extended to support other existing or future datasets. It can also be re-purposed for other face related tasks.

The work presented on this thesis can serve as a stepping stone for future age estimation studies. The synthetic dataset generation section of this thesis was done out of necessity. In the future, the described studies, specially the Renderings and Multi-View model, should be replicated using real life information and based in our observations, higher quality data should translate into better estimation performance.

The age estimation task itself can be improved with more complex problem formulations such as Ordinal Regression [NZW+16] or distance based projections of the face into an embedded space [HHM+17], both explored in Section 3. Although these approaches are not used in this project, they can be integrated into this thesis's work and we should expect similar improvements as the ones reported on their respective studies.

References

[AAB$^+$15] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[ABBD16] G. Antipov, M. Baccouche, S. A. Berrani, and J. L. Dugelay. Apparent age estimation from face images combining general and children-specialized deep learning models. In *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 801–809, June 2016.

[ASE17] A. Antoniou, A. Storkey, and H. Edwards. Data Augmentation Generative Adversarial Networks. *ArXiv e-prints*, November 2017.

[BBM12] A. D. Bagdanov, A. Del Bimbo, and I. Masi. Florence faces: A dataset supporting 2d/3d face recognition. In *2012 5th International Symposium on Communications, Control and Signal Processing*, pages 1–6, May 2012.

[Bra00] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[BRZ$^+$16] J. Booth, A. Roussos, S. Zafeiriou, A. Ponniahy, and D. Dunaway. A 3d morphable model learnt from 10,000 faces. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5543–5552, June 2016.

[BT17a] Adrian Bulat and Georgios Tzimiropoulos. Binarized convolutional landmark localizers for human pose estimation and face alignment with limited resources. *CoRR*, abs/1703.00862, 2017.

[BT17b] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). In *International Conference on Computer Vision*, 2017.

[BV99] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, pages 187–194, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[BW16] Geoff Boeing and Paul Waddell. New insights into rental housing markets across the united states: Web scraping and analyzing craigslist rental listings, 07 2016.

[C+15] François Chollet et al. Keras. https://keras.io, 2015.

[CCH11] K.-Y. Chang, C.-S. Chen, and Y.-P. Hung. Ordinal hyperplanes ranker with cost sensitivities for age estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[CET01] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, Jun 2001.

[CLL+15] Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. *CoRR*, abs/1512.01274, 2015.

[Csá01] Balázs Csanád Csáji. Approximation with artificial neural networks. 2001.

[CTCG95] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham. Active shape models-their training and application. *Computer Vision and Image Understanding*, 61(1):38 – 59, 1995.

[CTH+18] Feng-Ju Chang, Anh Tuan Tran, Tal Hassner, Iacopo Masi, Ram Nevatia, and Gérard G. Medioni. Expnet: Landmark-free, deep, 3d facial expressions. *CoRR*, abs/1802.00542, 2018.

[CV95] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, Sep 1995.

[CWZ+14] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou. Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):413–425, March 2014.

[CZD+17] S. Chen, C. Zhang, M. Dong, J. Le, and M. Rao. Using ranking-cnn for age estimation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 742–751, July 2017.

[DDS+09] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[DHS11] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12:2121–2159, July 2011.

[EEH14] E. Eidinger, R. Enbar, and T. Hassner. Age and gender estimation of unfiltered faces. *IEEE Transactions on Information Forensics and Security*, 9(12):2170–2179, Dec 2014.

[EFP+15] S. Escalera, J. Fabian, P. Pardo, X. Baró, J. Gonzàlez, H. J. Escalante, D. Misevic, U. Steiner, and I. Guyon. Chalearn looking at people 2015: Apparent age and cultural event recognition datasets and results. In *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, pages 243–251, Dec 2015.

[FKA+18a] Maayan Frid-Adar, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan. Synthetic data augmentation using GAN for improved liver lesion classification. *CoRR*, abs/1801.02385, 2018.

[FKA+18b] Maayan Frid-Adar, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan. Synthetic data augmentation using GAN for improved liver lesion classification. *CoRR*, abs/1801.02385, 2018.

[FM82] Kunihiko Fukushima and Sei Miyake. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In Shun-ichi Amari and Michael A. Arbib, editors, *Competition and Cooperation in Neural Nets*, pages 267–285, Berlin, Heidelberg, 1982. Springer Berlin Heidelberg.

[F.R01] Karl Pearson F.R.S. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[FWS+18] Yao Feng, Fan Wu, Xiaohu Shao, Yanfeng Wang, and Xi Zhou. Joint 3d face reconstruction and dense alignment with position map regression network. *CoRR*, abs/1803.07835, 2018.

[GA09] Feng Gao and Haizhou Ai. Face age classification on consumer images with gabor feature and fuzzy lda method. In Massimo Tistarelli and Mark S. Nixon, editors, *Advances in Biometrics*, pages 132–141, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[GFB+17] Thomas Gerig, Andreas Forster, Clemens Blumer, Bernhard Egger, Marcel Lüthi, Sandro Schönborn, and Thomas Vetter. Morphable face models - an open framework. *CoRR*, abs/1709.08398, 2017.

[GMC+08] Ralph Gross, Iain Matthews, Jeffrey Cohn, Takeo Kanade, and Simon Baker. Multipie. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*. IEEE Computer Society, September 2008.

[GMFH09] G. Guo, Guowang Mu, Y. Fu, and T. S. Huang. Human age estimation using bioinspired features. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 112–119, June 2009.

[GN08] A. Gunay and V. V. Nabiyev. Automatic age classification with lbp. In *2008 23rd International Symposium on Computer and Information Sciences*, pages 1–4, Oct 2008.

[GPM+14] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Networks. *ArXiv e-prints*, June 2014.

[GXX+16] Bin-Bin Gao, Chao Xing, Chen-Wei Xie, Jianxin Wu, and Xin Geng. Deep label distribution learning with label ambiguity. *CoRR*, abs/1611.01731, 2016.

[GZSM07] X. Geng, Z. H. Zhou, and K. Smith-Miles. Automatic age estimation based on facial aging patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2234–2240, Dec 2007.

[HHM$^+$17] Y. He, M. Huang, Q. Miao, H. Guo, and J. Wang. Deep embedding network for robust age estimation. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 1092–1096, Sept 2017.

[HHT$^+$16] P Huber, G Hu, R Tena, P Mortazavian, P Koppen, WJ Christmas, M Ratsch, and J Kittler. A multiresolution 3d morphable face model and fitting framework. In *11th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, February 2016. Paper accepted for presentation at 11th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, 27-29 February 2016. Full text may be available at a later date.

[HKL94] Young Ho Kwon and Niels Lobo. Age classification from facial images. 74:1–21, 01 1994.

[HZRS15a] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[HZRS15b] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, abs/1502.01852, 2015.

[JBAT17] Aaron S Jackson, Adrian Bulat, Vasileios Argyriou, and Georgios Tzimiropoulos. Large pose 3d face reconstruction from a single image via direct volumetric cnn regression. *International Conference on Computer Vision*, 2017.

[Jia13] Yangqing Jia. Caffe: An open source convolutional architecture for fast feature embedding, 2013.

[Kar18] Andrej Karpathy. Stanford University CS231n: Convolutional Neural Networks for Visual Recognition. 2018.

[KB14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[Kin09] Davis E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009.

[KS14] V. Kazemi and J. Sullivan. One millisecond face alignment with an ensemble of regression trees. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1867–1874, June 2014.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[Lan04] Andreas Lanitis. The fg-net aging data base. http://sting.cycollege.ac.cy/~alanitis/fgnetaging/index.htm, 2004.

[LAR03] Fei-Fei Li, Marco Andreetto, and Marc 'Aurelio Ranzato. Caltech101 image dataset. 2003.

[LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436 EP –, May 2015.

[LHBB99] Yann LeCun, Patrick Haffner, Léon Bottou, and Yoshua Bengio. Object recognition with gradient-based learning. In *Shape, Contour and Grouping in Computer Vision*, page 319, 1999.

[LRBS09] K. Luu, K. Ricanek, T. D. Bui, and C. Y. Suen. Age estimation using active appearance models and support vector machine regression. In *2009 IEEE 3rd International Conference on Biometrics: Theory, Applications, and Systems*, pages 1–5, Sept 2009.

[LTC97] A. Lanitis, C. J. Taylor, and T. F. Cootes. Automatic interpretation and coding of face images using flexible models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):743–756, Jul 1997.

[MAE16] R. C. Malli, M. Aygün, and H. K. Ekenel. Apparent age estimation using ensemble of deep learning models. In *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 714–721, June 2016.

[MBPVG14] Markus Mathias, Rodrigo Benenson, Marco Pedersoli, and Luc Van Gool. Face detection without bells and whistles. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 720–735, Cham, 2014. Springer International Publishing.

[MCCD13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[MW12] Bryan Mendelson and Chin-Ho Wong. Changes in the facial skeleton with aging: Implications and clinical applications in facial rejuvenation. *Aesthetic Plast Surg*, 36(4):753–760, Aug 2012. 9904[PII].

[NH10] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pages 807–814, USA, 2010. Omnipress.

[Nie15] M.A. Nielsen. *Neural Networks and Deep Learning*. Determination Press, 2015.

[NYD16] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. *CoRR*, abs/1603.06937, 2016.

[NZW$^+$16] Z. Niu, M. Zhou, L. Wang, X. Gao, and G. Hua. Ordinal regression with multiple output cnn for age estimation. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4920–4928, June 2016.

[OAE16] G. Ozbulak, Y. Aytar, and H. K. Ekenel. How transferable are cnn-based features for age and gender classification? In *2016 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–6, Sept 2016.

[Ots79] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.

[Par74] Frederic Ira Parke. *A Parametric Model for Human Faces.* PhD thesis, 1974. AAI7508697.

[PGC+17] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[PLTC16] G. Panis, A. Lanitis, N. Tsapatsoulis, and T. F. Cootes. Overview of research on facial ageing using the fg-net ageing database. *IET Biometrics*, 5(2):37–46, 2016.

[PLTFC15] Gabriel Panis, Andreas Lanitis, Nicholas Tsapatsoulis, and Timothy F. Cootes. An overview of research on facial aging using the fg-net aging database. 5, 05 2015.

[PVG+11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[PVZ15] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *British Machine Vision Conference*, 2015.

[QMB17] Zakariya Qawaqneh, Arafat Abu Mallouh, and Buket D. Barkana. Deep convolutional neural network for age estimation based on vgg-face model. *CoRR*, abs/1709.01664, 2017.

[RTG15] 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.

[Rud17] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *CoRR*, abs/1706.05098, 2017.

[SA16] Frank Seide and Amit Agarwal. Cntk: Microsoft's open-source deep-learning toolkit. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 2135–2135, New York, NY, USA, 2016. ACM.

[SAD+08] Arman Savran, Neşe Alyüz, Hamdi Dibeklioğlu, Oya Çeliktutan, Berk Gökberk, Bülent Sankur, and Lale Akarun. Bosphorus database for 3d face analysis. In Ben Schouten, Niels Christian Juul, Andrzej Drygajlo, and Massimo Tistarelli, editors, *Biometrics and Identity Management*, pages 47–56, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[SAG+16] C. Sagonas, E. Antonakos, Tzimiropoulos G., S. Zafeiriou, and M. Pantic. *300 faces In-the-wild challenge: Database and results.* Image and Vision Computing (IMAVIS), Special Issue on Facial Landmark Localisation "In-The-Wild", 2016.

[SKP15] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. *CoRR*, abs/1503.03832, 2015.

[SMDH13] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1139–1147, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

[SMKL15] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik G. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. *CoRR*, abs/1505.00880, 2015.

[Sob14] Irwin Sobel. An isotropic 3x3 image gradient operator. 02 2014.

[STZP13] Christos Sagonas, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. A semi-automatic methodology for facial landmark annotation. In *2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPRW)*, pages 896–903, June 2013. © 2013 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

[SWT13] Y. Sun, X. Wang, and X. Tang. Deep convolutional network cascade for facial point detection. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3476–3483, June 2013.

[SZ14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[TCT16] Qing Tian, Songcan Chen, and Xiaoyang Tan. A unified gender-aware age estimation. *CoRR*, abs/1609.03815, 2016.

[THM+17] Anh Tuan Tran, Tal Hassner, Iacopo Masi, Eran Paz, Yuval Nirkin, and Gérard G. Medioni. Extreme 3d face reconstruction: Looking past occlusions. *CoRR*, abs/1712.05083, 2017.

[THMM16] Anh Tuan Tran, Tal Hassner, Iacopo Masi, and Gérard G. Medioni. Regressing robust and discriminative 3d morphable models with a very deep neural network. *CoRR*, abs/1612.04904, 2016.

[TN17] Luke Taylor and Geoff Nitschke. Improving deep learning using generic data augmentation. *CoRR*, abs/1708.06020, 2017.

[TPA+18] Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, and Stan Birchfield. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. *CoRR*, abs/1804.06516, 2018.

[VBF11] V. Vijayan, K. Bowyer, and P. Flynn. 3d twins and expression challenge. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 2100–2105, Nov 2011.

[WGK15] X. Wang, R. Guo, and C. Kambhamettu. Deeply-learned feature for age estimation. In *2015 IEEE Winter Conference on Applications of Computer Vision*, pages 534–541, Jan 2015.

[XWTQ04] Chenghua Xu, Yunhong Wang, Tieniu Tan, and Long Quan. Automatic 3d face recognition combining global geometric features with local shape variation information. In *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings.*, pages 308–313, May 2004.

[YCBL14] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *CoRR*, abs/1411.1792, 2014.

[YLL15] Dong Yi, Zhen Lei, and Stan Z. Li. Age estimation by multi-scale convolutional network. In Daniel Cremers, Ian Reid, Hideo Saito, and Ming-Hsuan Yang, editors, *Computer Vision – ACCV 2014*, pages 144–158, Cham, 2015. Springer International Publishing.

[YLW+14] Chenjing Yan, Congyan Lang, Tao Wang, Xuetao Du, and Chen Zhang. Age estimation based on convolutional neural network. In Wei Tsang Ooi, Cees G. M. Snoek, Hung Khoon Tan, Chin-Kuan Ho, Benoit Huet, and Chong-Wah Ngo, editors, *Advances in Multimedia Information Processing – PCM 2014*, pages 211–220, Cham, 2014. Springer International Publishing.

[YWS+06] Lijun Yin, Xiaozhou Wei, Yi Sun, Jun Wang, and M. J. Rosato. A 3d facial expression database for facial behavior research. In *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, pages 211–216, April 2006.

[ZDH17] T. Zheng, W. Deng, and J. Hu. Age estimation guided convolutional neural network for age-invariant face recognition. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 503–511, July 2017.

[ZGG+17] Ke Zhang, Ce Gao, Liru Guo, Miao Sun, Xingfang Yuan, Tony X. Han, Zhenbing Zhao, and Baogang Li. Age group and gender estimation in the wild with deep ror architecture. *CoRR*, abs/1710.02985, 2017.

[ZLY+18] K. Zhang, N. Liu, X. Yuan, X. Guo, C. Gao, and Z. Zhao. Fine-Grained Age Estimation in the wild with Attention LSTM Networks. *ArXiv e-prints*, May 2018.

[ZXKJ+17] Jian Zhao, Lin Xiong, Panasonic Karlekar Jayashree, Jianshu Li, Fang Zhao, Zhecan Wang, Panasonic Sugiri Pranata, Panasonic Shengmei Shen, Shuicheng Yan, and Jiashi Feng. Dual-agent gans for photorealistic and identity preserving profile face synthesis. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 66–76. Curran Associates, Inc., 2017.

Appendix A

Model Architectures

In this section, we will illustrate the models architectures.

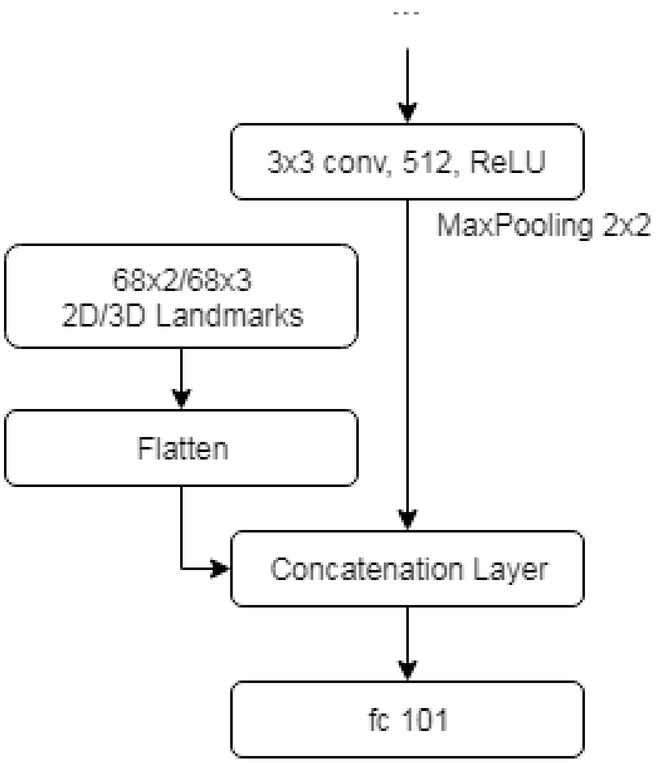


Figure A.1: The architecture of the Landmark Model. "..." indicate the rest of the model, as it is based on the baseline model.

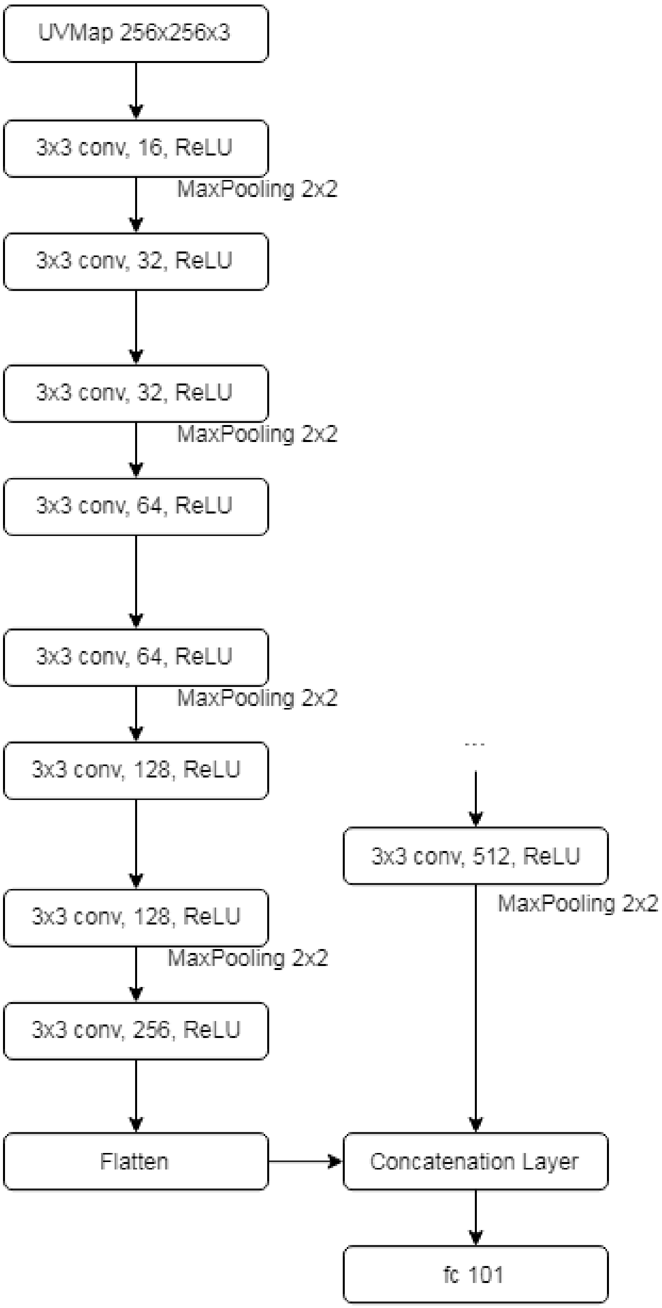


Figure A.2: The architecture of the U.V. Maps Model. "..." indicate the rest of the model, as it is based on the baseline model.

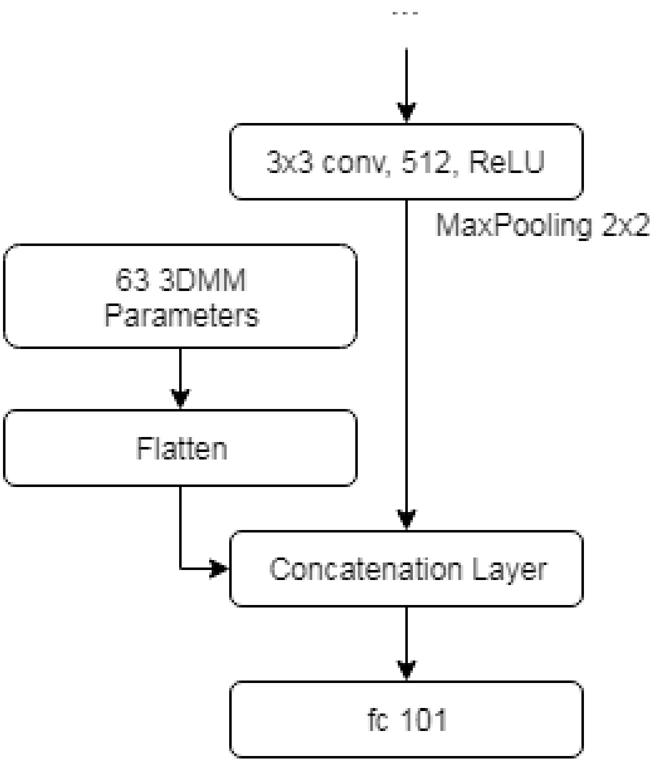


Figure A.3: The architecture of the 3DMM Model. "..." indicate the rest of the model, as it is based on the baseline model.

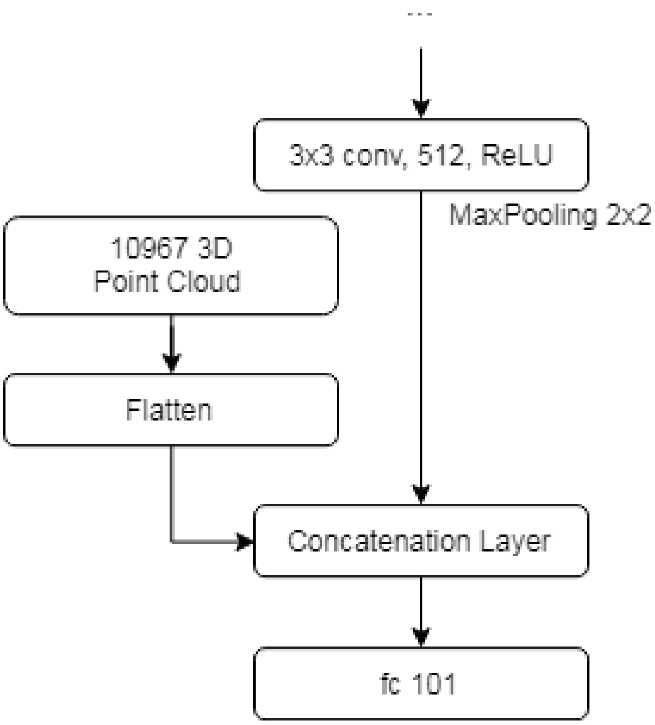
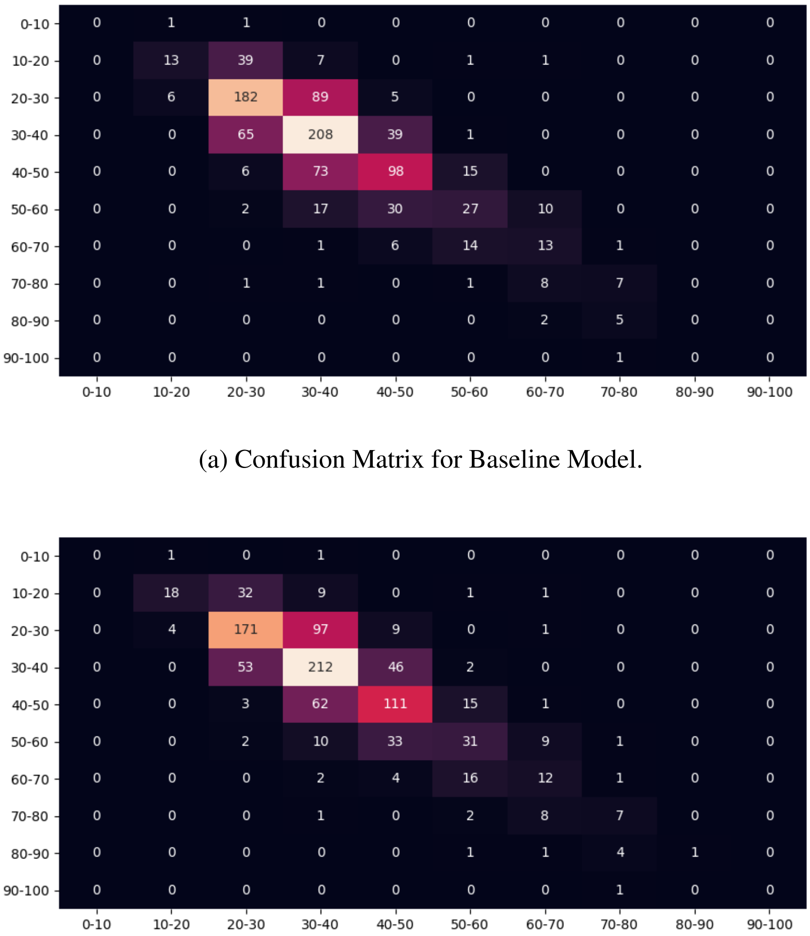


Figure A.4: The architecture of the Point Cloud Model. "..." indicate the rest of the model, as it is based on the baseline model.

Appendix B

Pre-Defined Bags

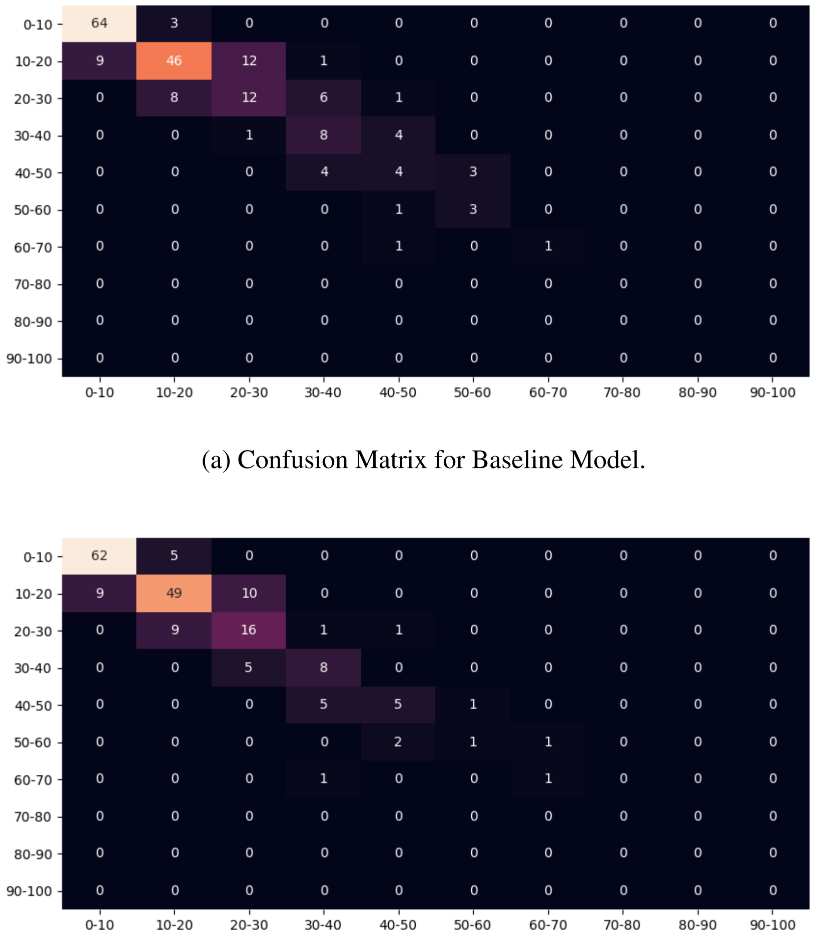


(a) Confusion Matrix for Baseline Model.

(b) Confusion Matrix for Renderings Model.

Figure B.1: Confusion Matrices on IMDB-WIKI dataset (Pre-defined bags).

Pre-Defined Bags

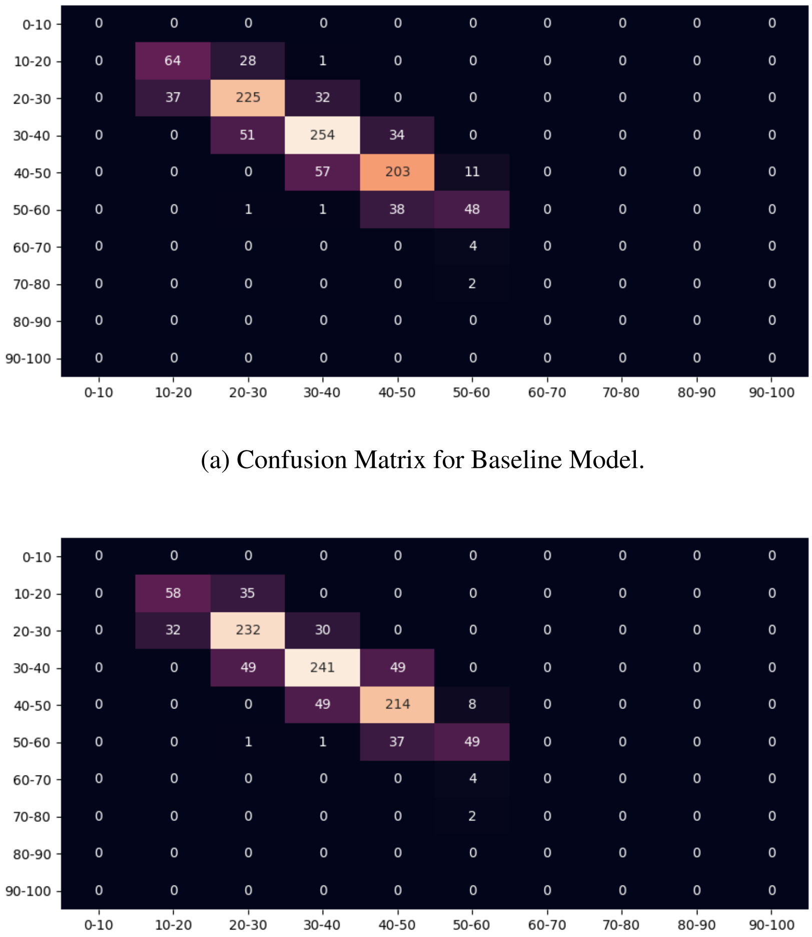


(a) Confusion Matrix for Baseline Model.

(b) Confusion Matrix for Renderings Model.

Figure B.2: Confusion Matrices on FG-NET-AD dataset (Pre-defined bags).

Pre-Defined Bags



	0-10	10-20	20-30	30-40	40-50	50-60	60-70	70-80	80-90	90-100
0-10	0	0	0	0	0	0	0	0	0	0
10-20	0	64	28	1	0	0	0	0	0	0
20-30	0	37	225	32	0	0	0	0	0	0
30-40	0	0	51	254	34	0	0	0	0	0
40-50	0	0	0	57	203	11	0	0	0	0
50-60	0	0	1	1	38	48	0	0	0	0
60-70	0	0	0	0	0	4	0	0	0	0
70-80	0	0	0	0	0	2	0	0	0	0
80-90	0	0	0	0	0	0	0	0	0	0
90-100	0	0	0	0	0	0	0	0	0	0

(a) Confusion Matrix for Baseline Model.

	0-10	10-20	20-30	30-40	40-50	50-60	60-70	70-80	80-90	90-100
0-10	0	0	0	0	0	0	0	0	0	0
10-20	0	58	35	0	0	0	0	0	0	0
20-30	0	32	232	30	0	0	0	0	0	0
30-40	0	0	49	241	49	0	0	0	0	0
40-50	0	0	0	49	214	8	0	0	0	0
50-60	0	0	1	1	37	49	0	0	0	0
60-70	0	0	0	0	0	4	0	0	0	0
70-80	0	0	0	0	0	2	0	0	0	0
80-90	0	0	0	0	0	0	0	0	0	0
90-100	0	0	0	0	0	0	0	0	0	0

(b) Confusion Matrix for Renderings Model.

Figure B.3: Confusion Matrices on MORPH2 dataset (Pre-defined bags).