MASTER'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Hroch Jan** Personal ID number: **458012**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Computer Science**

Study program: **Open Informatics**

Specialisation: **Artificial Intelligence**

II. Master's thesis details

Master's thesis title in English:

Large database of annotated face sequences

Master's thesis title in Czech:

Velká databáze anotovaných sekvencí tváří

Guidelines:

The goal of this thesis is to create a large database of human face sequences extracted from video which will be annotated by attributes like age, gender and identity. The source data will be movie trailers downloaded from Internet along with open-source personal data and photographs of actors staring in the movies. The task will be to develop a method that will learn appearance models of the actors from the photographs and use the models to link faces tracked in the trailers with the actor's personal data. The method should require minimal human intervention and at the same time it should provide as accurate annotation as possible. The important task will be to reliably estimate accuracy of the automatically collected annotation.
Tasks:
1. Provide a literature survey on methods for automated face annotation
2. Implement scripts for automated downloading movie trailers and associated actors' data
3. Develop a method for tracking and annotation of faces in the movie trailers
4. Evaluation accuracy of the created annotated database

Bibliography / sources:

[1] R. Rothe, R. Timofte, L.V. Gool. Dex: Deep expectation of apparent age from a single image. In IEEE International Conference on Computer Vision Workshops. 2015.
[2] E. Ghaleb, M. Tapaswi, Z. Al-Halah. Accio: A data set for face track retrieval in movies across age. In Proc. of ICMR, 2015.
[3] V. Franc, J. Cech. Learning CNNs from Weakly Annotated Facial Images. Journal of Image and Vision Computing, 2018.

© ČVUT v Praze, Design: ČVUT v Praze, VIC

Name and workplace of master's thesis supervisor:

Ing. Vojtěch Franc, Ph.D., Machine Learning, FEE

Name and workplace of second master's thesis supervisor or consultant:

Date of master's thesis assignment: **04.02.2020** Deadline for master's thesis submission: **22.05.2020**

Assignment valid until: **30.09.2021**

Ing. Vojtěch Franc, Ph.D. Supervisor's signature	Head of department's signature	prof. Mgr. Petr Páta, Ph.D. Dean's signature

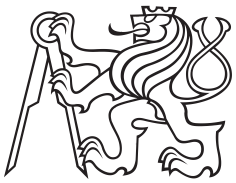
III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master's thesis, the author must state the names of consultants and include a list of references.

.	
Date of assignment receipt	Student's signature

© ČVUT v Praze, Design: ČVUT v Praze, VIC

Master Thesis



Czech Technical University in Prague

F3

Faculty of Electrical Engineering
Department of Computer Science

Large database of annotated face sequences

Bc. Jan Hroch

This work was supported by the Czech Science Foundation Project GAČR GA19-21198S
Supervisor: Ing. Vojtěch Franc, Ph.D.
May 2020

Acknowledgements

I wish to express my gratitude to my supervisor, Mr. Franc, for his valuable advice and mentorship. The help with the manual annotation of database used for testing was also crucial. Therefore, I would like thank everyone that contributed.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis. I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular that the Czech Technical University in Prague has the right to conclude a license agreement on the utilization of this thesis as a school work under the provisions of Article 60(1) of the Act.

In Prague, 22. May 2020

. . . . . .

Abstract

The thesis proposes a method to automatically annotate face sequences found in movie trailers. The method was applied on trailers downloaded from Internet Movie Database (`www.imdb.com`). The resulting database, called IMDB video faces, contains 14,000 face sequences of celebrities annotated by age, gender and identity. In addition, we created a small-size database of 6,619 face sequences with ground-truth annotation created manually. The manually annotated database was used to tune parameters of the proposed algorithm and to evaluate accuracy with which the annotations are assigned to tracks.

Keywords: Annotated face sequences, Age and gender prediction

Supervisor: Ing. Vojtěch Franc, Ph.D.

Abstrakt

Tato práce navrhuje metodu na automatické anotování sekvencí tváří nalezených v upoutávkách k filmům. Tato metoda byla aplikovaná na filmové upovtávky stažené z Internet Movie Database (`www.imdb.com`). Výsledná databáze, nazvaná IMDB video faces, obsahuje 14 000 sekvencí tváří celebrit anotované věkem, pohlavím a identitou. Kromě toho jsme vytvořili malou databázi s 6 619 sekvencemi tváři s manuálně vytvořenou anotací. Tato lidmi anotovaná databáze byla použita k vyladění parametrů navrhovaného algoritmu a k vyhodnocení přesnosti, se kterou jsou anotace přiřazeny k sekvencím tváří.

Klíčová slova: Anotované sekvence tváří, Detekce věku a pohlaví

Překlad názvu: Velká databáze anotovaných sekvencí tváří

Contents

Figures

Tables

Chapter 1

Introduction

Predicting age and gender from images is a long standing computer vision problem. It is not only theoretical problem but it also has multiple applications. For example, it has been used for demographic surveys, personalized advertisement and as a sub-system of age-invariant face recognition. Imagine advertisement boards on a street that are able to promote products based on age and gender of people standing around. To create such systems it is important to have as accurate predictions as possible.

The current state-of-the-art predictors of age and gender are based on convolutional neural networks learned from data by supervised methods [1, 2, 3, 4, 5, 6]. These methods require database of images annotated by target variables, that is, by age and gender. The databases are not only necessary for training the predictor but also for its reliable evaluation. Age/gender prediction methods studied so far use still images as the input modality. Therefore most of existing public databases contain annotated still images (c.f. Table 2.1). For example, the IMDB-WIKI database [7] is the largest age/gender database that was created by an automated algorithm processing data downloaded from Internet Movie Database (`www.imdb.com`).

Age/gender prediction from videos has been largely overlooked so far, despite the fact that the most applications use video as the input. Reason for low activity in this field is arguably lack of databases. Goal of this thesis is to fill this gap by creating a large database of face sequences (tracks) annotated by age, gender and identity. In particular, we created a database similar to IMDB-WIKI with the difference that it contains annotated sequences of face images found in videos instead of annotated still images. Using ideas similar to [7], we are able to compute age of celebrities shown in movie trailer using the movie release year and celebrity birth year along with information of his/her gender and name. The new challenge is to link this information (per-se associated with a trailer) to face tracks automatically found in the trailers. To this end, we propose an approach which can create the database of annotated video tracks in fully automated way and hence it is applicable to large data. To tune the parameters of proposed algorithm we also created a small-size database of tracks which is annotated manually in application we developed for that purpose. The manually annotated database is also exploited to estimate accuracy with which the proposed algorithm assigns

annotation to tracks. Finally, the proposed method is used to create a database of annotated video tracks which is to our knowledge the largest database of this kind.

The thesis is organized as follows. Chapter 2 provides a brief overview of related works. Chapter 3 describes the proposed automatic annotation algorithm. A process behind creation of the manually annotated database is discussed in Chapter 4. Tuning and evaluation of the proposed annotation algorithm on the manually created database is a subject of Chapter 5. Finally, Chapter 6 describes the resulting database, called IMDB video faces, containing image sequences annotated with age, gender and identity.

Chapter 2

State of the art

Machine learning methods applied for face recognition have been powered by data. This thesis focuses on databases for age and gender prediction from videos. Research in this area has been centered around prediction from still facial images while videos as the input modality has been largely neglected so far as witnessed by existing databases summarised in Table 2.1. Though not aspiring to be exhaustive it contains most of publicly accessible databases appearing in scientific literature.

Accio database [8] contains face tracks extracted from Harry Potter movies. Automatically extracted 38,364 face tracks are manually annotated by one out of 121 character identities and roughly 40% of tracks are marked as unknown character. Face tracks are assigned chronological age of captured identities. The age is computed based on actor's birth year and year of the movie release. The age span is from 10 to (approximately) 80 years. The database is primarily meant for studying age invariant identity recognition. Though it can be applied for age prediction as well its major disadvantage for this task is a relatively small number of subjects. Additional problem is a relatively limited variation as most of the captured tracks originate from dark scenes typical for the movie.

This thesis proposes method that automatically annotates video tracks extracted from movie trailers downloaded from IMDB website `www.imdb.com`. Data from IMDB website has been previously used to create IMDB-WIKI database [7] which is currently the largest public collection of still facial images of celebrities annotated by age and gender. The database has been used by many recent papers related to age prediction. The database was created by a fully automated process that exploits a creation time of the images and known birth date of captured identities. For each image, they compute a biological age of the respective celebrity by subtracting the birth date from the creation time. However, besides the celebrity most of the images contain other identities. Hence the challenge is to link the obtained age annotation to the correct face in the image. The authors of [7] use a simple heuristic which discards all images but those with a single face detection which is assumed to correspond to the celebrity. A statistically grounded approach to devise a clean annotation of IMDB images (Wikipedia images were excluded) was proposed in [9]. Their method describes appearance of the identities, their age

Still images

Dataset	Year	#Faces	#Subj
FG-NET [10]	2004	1,002	82
MORPH2 [11]	2006	55,000	13,000
GroupsDataset [12]	2009	28,231	N/A
Adience [13]	2014	26,580	2,284
CACD [14]	2014	163,446	2,000
IMDB-WIKI [7]	2015	523,051	20,284
ChaLearn [15]	2016	7,591	N/A
AFAD [3]	2016	165,501	N/A
AgeDB [16]	2017	16,488	568
AppaReal [17]	2017	7,591	7,000
UTKFace [18]	2017	20,000	N/A

Videos

Dataset	Year	#Tracks	#Subj
Accio [8]	2015	38,464	121

Table 2.1: Summary of major public face databases with age annotation.

and gender by a single statistical model parameters which are learned from examples by the maximum likelihood approach. The estimated model allows to assign the annotation to the detected faces with a prescribed confidence. The resulting annotation is more precise and the number of annotated faces is higher as the method discards the images with a single face detection. A downside of the method are relatively high computational requirements.

Similarly to [9], the method proposed in this thesis also uses appearance models of celebrities to link the annotation to the face tracks.

Chapter 3
Methods

The main goal of this thesis is to create a large database of face sequences annotated by age and gender. This chapter starts with a more detailed definition of the problem to be solved given in Section 3.1. It is followed by description of building blocks of the proposed approach to automatic annotation of face sequences extracted from movie trailers. Namely, Section 3.2 describes face detector used for finding faces in video frames, Section 3.3 characterizes how were detected faces connected into tracks, Section 3.4 describes the way we used face recognition and Section 3.5 specifies how we built appearance models for celebrities. Finally, Section 3.6 summarizes the whole process of automatic track annotation.

3.1 Problem definition

There is a huge amount of movie trailers available on IMDB (Internet Movie Database) website `www.imdb.com`. We are able to get a set of trailers annotated with age and gender of a celebrity acting in the movie. We are also able to get set of images for most of the celebrities. Those images can be split into two groups. Portraits which contain only selected celebrity and images showing celebrity along with other people. Using the images we have for each celebrity the goal is to find tracks (sequences of face images) in above-mentioned trailers belonging to the celebrity associated with the trailer. These tracks can be then annotated by the same age, gender and celebrity name as the trailer.

3.2 Face detection

We are using Python implementation [19] of MTCNN [20] to detect faces in images. This neural network is a face detector which predicts positions of faces in given image. For each detected face in an image it provides the following information:

- Bounding box described by position, width and height
- Confidence

- Position of facial key points:
  - Nose
  - Left and right eye
  - Left and right corner of mouth

3.3 Face tracker

Track is a sequence of faces found in video that correspond to continuous appearance of a single subject in a scene. We used *tracking by detection* approach to find face tracks. First we detect faces in a video frame. Each detected face is described by its bounding box. Then we continue to the next frame and find faces again. Now we match the faces found in frame $n + 1$ to faces detected in frame n. This is done by calculating the overlap of the face bounding boxes measured by the intersection over union of the bounding box areas. If the shared area takes more than 65% of the combined space we say that both faces belong to the same track. Basically we use the following condition to decide whether the face bounding boxes A and B belong to the same track.

$$IOU(A, B) = \frac{A \cap B}{A \cup B} > 0.65\,.$$

Sometimes the used face detector fails to detect face we are already tracking. It might be due to sudden lighting change in the video or by some other causes like face being partially covered by hand or other objects. To deal with this, we do not end track right after failing to match it to any face. In our implementation of face tracker the track is ended after 0.5 second of not matching a face to it. The actual number of frames depends on frame rate and it is usually between 12 to 15 frames. We also limited the minimal length of track to 5 images which helps to lower the amount of false positives by the face detector.

3.4 Face recognition

In order to describe identity of faces found by the face detector we are using VGGFace2 [21]. It is a neural network which for given face image outputs a feature vector containing information of the identity. The dimension of the feature vector varies between model architectures. For example the ResNet-50 architecture describes face by $2,048$ features while the SE-ResNet-50-256D architecture uses only 256 features. We experimented with both the architectures.

The input shape of an image for the neural network is 224×224px. Because the bounding box given by the face detector is not of a square shape we transform detected faces in the following way. First we resize the face image so that its shorter size is 224 pixels long and then crop the 224×224 center. This allows to use detected face images without changing their aspect ratio.

To measure how similar two identity feature vectors extracted from face images are we use the cosine distance. Given vectors u and v describing identity, the cosine distance is defined as:

$$D(u, v) = 1 - \frac{u \cdot v}{||u||_2 ||v||_2}$$

which is a value from interval $[0, 2]$. The lower value is the more similar the vectors and consequently the identities are.

To decide whether two images I_1 and I_2 belong to the same person or not we measure the cosine distance between their feature vectors $\phi(I_1)$ and $\phi(I_2)$ extracted by the neural network. This is done using the following approach:

$$\text{same\_identity}(I_1, I_2) = \begin{cases} yes, & \text{if } D(\phi(I_1), \phi(I_2)) < \Theta \\ no, & \text{otherwise} \end{cases}$$

where Θ is a distance threshold. The exact threshold value depends on the purpose the face recognition is used for. Imagine that, for instance, we want to have a low amount of false positives then we choose low threshold. To the contrary, by increasing the threshold we can decrease the number of false negatives.

The aforementioned method for deciding whether two images belong to the same person can be extended to two sets of images as opposed to just two images. Using multiple images to represent each person increases the robustness of this method because it allows us to capture person's appearance in multiple stages of their life. To identify a person we first need example images of their face. This set of images can be then transformed into a single identity descriptor. This identity descriptor is a vector which represents person's appearance. In our case, the other set of images with unknown identity is a track. There are multiple approaches to comparing identity descriptor with multiple images. We calculate distance for each face separately. Then we combine those distances into a single number by:

- calculating median,
- averaging.

Finally, we decide based on the calculated median/average of cosine distances and used threshold Θ similarly as in previous method. Using either median or average differs in the way they address changing face setting like pose, expression, lightning etc. for images in the same track.

To conclude, this method has following parameters:

- Face bounding box multiplier, i.e. by how much we increase the bounding box's size given by the face detector. This allows the face images to contain additional information, such as hair, beard or ears which influences the accuracy of face recognition.
- VGGFace2 architecture used to extract identity feature vectors.

- Approach used to compare identity descriptor of known person with track:
  - median
  - average

  of distances to each image in track.
- Threshold Θ

3.5 Appearance model

To recognize identity in an image we need to create its appearance model. We do this by taking face images that belong to the identity and extract feature vector from each face by VGG-Face2. To represent identity by a single vector we then use the extracted vectors to calculate either their coordinate-wise median or their average vector. Both of those approaches yield different results as we show in Chapter 5. We also call such representation of person's appearance the celebrity descriptor.

To build the celebrity descriptors we use the following two sources of images:

- Portraits found on TMDB website `www.themoviedb.org`. The portrait images contain a single face belonging to the celebrity.
- Pictures taken directly from movies or images shot at movie related events. These images can be found on IMDB website and may contain multiple persons. However, the target celebrity is assumed to be the most frequent occurring face.

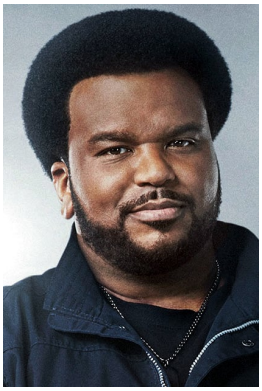
Method of building appearance models using the two sources is described in the following subsections.

3.5.1 TMDB portraits

The portrait images available on TMDB website have generally quite high resolution and contain only the chosen celebrity. An example of TMDB portrait images is shown in Figure 3.1. Usually, the celebrity is looking straight into the camera and the face covers substantial portion of the image. To create the celebrity descriptor from these images we simply have to find face in each image and extract its feature vector. The celebrity descriptor is then computed as either an average or a coordinate-wise median of the extracted vectors.

(a) : Jenny Slate

(b) : Andy Samberg



(c) : Craig Robinson

Figure 3.1: Example of TMDB portraits.

3.5.2 IMDB images

Using images found on IMDB website is more complicated. The images can be downloaded from a profile page dedicated to the celebrity. We have multiple images for each celebrity. However, these pictures, usually being photos from social events or frames captured from movies, may contain multiple people and the celebrity associated with the image may not even be any of them in some cases. Although the celebrity is in most of the photos we do not know which face belongs to him/her. To create an accurate celebrity descriptor from those images we first need to identify faces belonging to the celebrity.

As an example Figure 3.2 shows an image found on Steve Carell's IMDB profile page. The image contains multiple celebrities and we don't know which face belongs to Steve Carell.

Figure 3.2: Example IMDB image with multiple celebrities.

We can assume that the chance that a single identity is in an image more than once is very low. Sometimes it happens, for example when there are twins in a movie portrayed by the same actor as shown in Figure 3.3 or there

is a mirror in the movie scene, but it is very rare. We can use this to our advantage when filtering faces of the selected celebrity by selecting at most one face from each image.

Figure 3.3: Actor James Franco portraying twin brothers.

To separate faces found in images from celebrity's IMDB profile into those that belong to the celebrity and the rest we use the following method divided into 4 steps.

1. Find faces in all images.
2. Use VGG-Face2 to extract identity feature vector from each face and calculate coordinate-wise median from all vectors. This method was used e.g. in [9].
3. For each image, find a face with the lowest distance to the median vector from Step 2 and if the distance is lower than a threshold mark the face as belonging to the celebrity.
4. Create the celebrity descriptor either by averaging or by calculating coordinate-wise median of identity feature vectors extracted from faces marked in Step 3.

The median created in Step 2 represents the identity's appearance to some extent because the most frequent identity in those picture is the selected celebrity. We use this approximate identity model to further filter found faces.

The idea behind selecting only faces with distance lower than a set threshold is to discard faces that are closest to the median (from Step 2) but might not belong to the selected celebrity. For instance this is the case when the celebrity is in an image but his/her face is not visible. This occurs, for example, when the image shows the celebrity from behind or when the celebrity's face was overlooked by face detector. Selecting any face from such image is obviously incorrect.

Correctly separating faces belonging only to the selected celebrity is crucial for creating accurate appearance model. Because we do not have human annotations for those faces we cannot directly measure how well we are able

to classify them. The accuracy relies heavily on the threshold used in Step 3. Tuning the threshold is described in Section 5.5.1

3.5.3 Method parameters

To conclude, the process of creating celebrity descriptors has the following parameters:

- Face bounding box multiplier
- VGGFace2 architecture
- Images used to create the celebrity descriptor:
  - TMDB portraits
  - IMDB images
    - Threshold θ used for filtering celebrity's face images
- Approach used to merge extracted identity feature vectors into celebrity descriptor:
  - average
  - median

3.6 Procedure for automated IMDB video annotation

An automated procedure for annotating large set of video tracks, being the main objective of the thesis, is described in this section. It uses building blocks described above. The annotation process works as follows:

1. Select subset of celebrities.
2. Scan celebrities' IMDB profile pages for pairs of trailers and related movies.
3. Calculate age of celebrity for each pair of trailer and related movie using the celebrity's birth year and the movie's release year both obtained from datasets provided by IMDB.
4. Select subset of trailers.
5. Build celebrity descriptors for celebrities appearing in selected trailers which was described in Section 3.5.
6. Find tracks in selected videos.
7. For each trailer, find tracks which belong to the target celebrity based on the distance between the track and the celebrity descriptor. The approach was described in Section 3.4.

8. Assign age, gender and identity to tracks found in Step 7 using the information from Step 3.

Chapter 5 focuses on tuning the parameters of building blocks used in the proposed annotation approach.

3.7 Noise in annotated attributes

One of the disadvantages of used annotation approach is the delay between filming a movie and its release. On average, the delay between the day shooting begins and the movie's release date is 407 days [22]. Another drawback is that some videos are compilations of multiple movies. For instance documentaries describing celebrity's life. Such trailers contain movie scenes from multiple movies but the release year of the documentary does not match the release year of those movies. This leads to incorrect age annotations for tracks found in such videos. Fortunately for us, such videos are rare.

3.8 Evaluation metrics

To measure the accuracy of proposed annotation algorithm with various parameter configurations we will use manually annotated database of tracks described in Chapter 4 and evaluate the following attributes:

- **percentage of incorrectly annotated tracks**,
- **extraction percentage**, i.e. how many of the tracks belonging to the celebrity (that is supposed to be in the trailer) we were able to annotate.

Values of both metrics are directly tied to threshold Θ used when annotating tracks. These metrics are somewhat contradictory. Selecting threshold to reach low percentage of incorrectly annotated tracks will mean throwing away a large portion of tracks belonging to the target celebrity. Chapter 5 offers evaluation of introduced metrics for a range of thresholds Θ to provide more complex analysis of various parameter configurations.

Chapter 4

Manually annotated database

To tune hyper-parameters and to evaluate annotation error of the proposed annotation algorithm we created database which was annotated manually. This database was created from a small subset of IMDB videos. We chose 50 popular actors and 50 popular actresses for this database. Creation of the manually annotated database is a subject of this chapter.

4.1 Trailer selection

Our goal was to create a database of video tracks with a balanced age and gender distribution. To achieve this goal we selected 500 trailers with actors and 500 trailers with actresses while trying to keep the age distribution as even as possible. The algorithm used to select these trailers is similar to Algorithm 1. In order to create an even distribution we limited the maximum number of trailers in each age group to 10.

Figure 4.1 shows the age distribution of celebrities starring in selected trailers. It is observable that the selection is not perfectly even. This may be caused by the fact that only very few of those 50 selected actresses are older. To be precise, none of the selected actresses starred in a movie where their age was greater than 74 years. It can be also seen that older actors are more popular than older actresses.

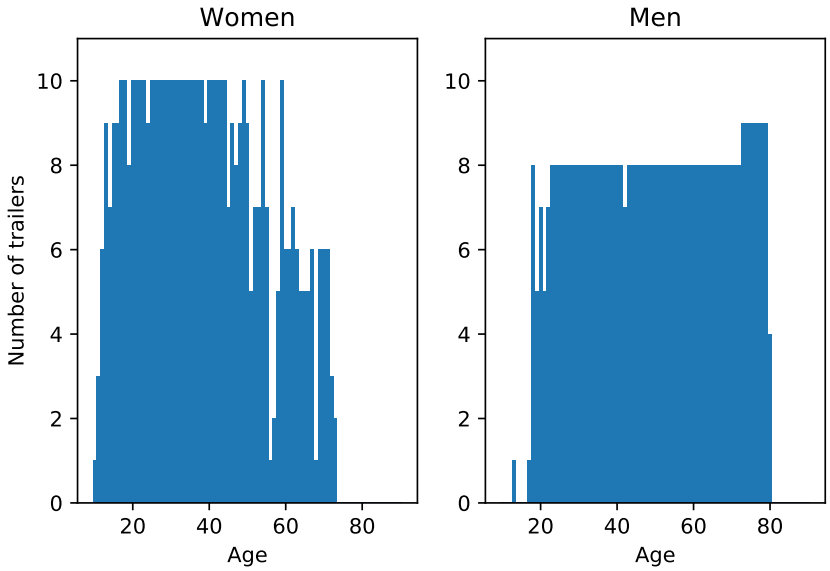


Figure 4.1: Age distribution of celebrities in selected trailers.

Another goal was to also have distribution of the number of trailers per identity as even as possible. It is not ideal to have great portion of tracks belonging to one celebrity. We limited the maximum number of trailers per celebrity to 10. But in order to fulfill the previous goal some exceptions had to be made.

Figure 4.2 describes how many trailers per celebrity were selected. The figure shows that we selected 80 trailers starring actress Susan Sarandon. As mentioned earlier, this is not perfect. On the other hand, Mrs. Sarandon has many trailers in age groups in which other actresses do not. If we had not made this exception there would be less trailers for women in age group from 40 to 73 years old. It is clear from the figure that a few other exceptions were made for the same reason for both men and women.

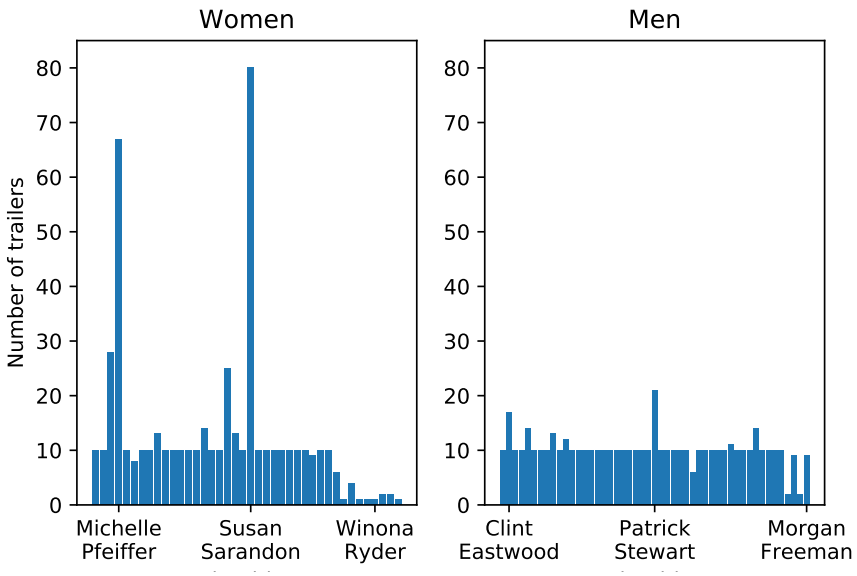


Figure 4.2: Number of trailers per identity.

We ended up with 89 celebrities with at least one selected trailer where 49 of them are actors and 40 are actresses.

4.2 Annotation tool

We created a simple tool that allows to manually annotate tracks found in selected videos. The purpose of this section is to describe how were the tracks annotated which should give readers an idea how accurate the human annotations are.

Figure 4.3 shows a snapshot of web application we developed to annotate found tracks. The web page displays a table where each row corresponds to one track extracted from given trailer. Each track is summarized by 5 facial images evenly selected from the beginning to the end of the track. Each track is described by its distance to the celebrity's appearance model. The celebrity descriptors used to calculate those distances were created utilizing the median of feature vectors extracted from TMDB portraits. The calculated distances are used to sort the tracks in the table in an ascending order, i.e. tracks most likely belonging to the target celebrity are shown first. We believe that this helps to speed up the annotation process and also improve accuracy of annotators. That's because the lower the distance, the higher the probability that the track belongs to the selected celebrity. The sorting of tracks can help the annotators to focus their attention on the borderline cases instead of inspecting each track independently.

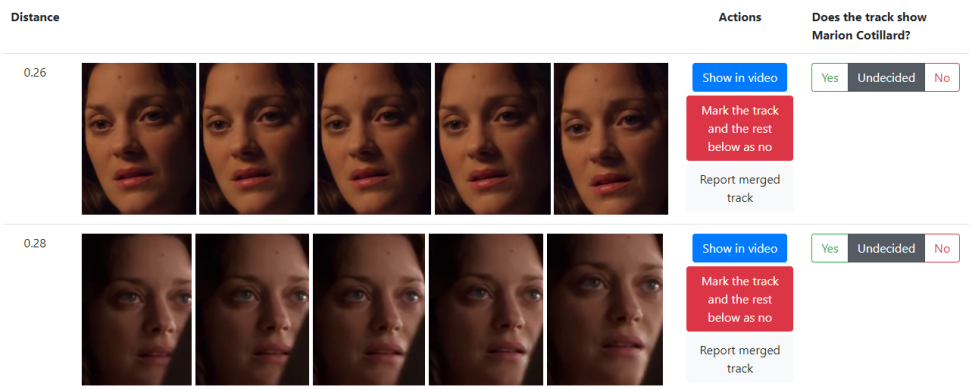


Figure 4.3: A snapshot of the web application we developed to manually annotate face tracks extracted from trailers of a target celebrity. The example shows tracks found in a trailer starring actress Marion Cotillard, being here the target celebrity. The annotator is asked to mark which tracks belong to the celebrity and which do not. Besides the 5 image summary, the annotator has an option to watch whole track in source video.

The annotator is asked to decide whether each track captures the selected celebrity or not. In addition, each track can be also reported as "merged" which means that it contains multiple identities due to a failure of the face tracker. This happens when there is a cut in the video and the following scene contains a face around the position of the face in previous scene. This is more common for trailers compared to movies, because trailers usually consist of many scenes in short amount of time thus contain a large amount of cuts. We measured the number of merged tracks reported by annotators and it was relatively low. The number of 234 tracks were reported as merged which corresponds to around 0.43% of all tracks.

Additional feature of the annotation tool is that it allows to re-play part of the video where the track to be annotated occurs. This provides more information needed for annotation than just the 5 image summary. Namely, seeing the whole scene may help to better associate faces with the characters. For example, when a face is of very low quality but we may see that it belongs to a character wearing white coat and red hat. Let us say that one of the tracks annotated previously featured a character in the same clothes. Associating those tracks together can help the annotator to reach a more informed decision. A snapshot demonstrating this feature is shown in Figure 4.4. Faces marked by green rectangle are those which belong to the track that is to be annotated. The faces from other tracks are distinguished by red rectangles.

Figure 4.4: Annotation tool showing selected track in a video.

4.3 Annotated tracks

The total number of found tracks in the selected trailers is 53,994. Out of this number 6,619(12.3%) tracks capture the target celebrities with known identity. The remaining tracks correspond to unknown subjects appearing in the same movie.

For each track we computed an average size of bounding boxes of faces in the track. Figure 4.5 shows a histogram of the computed average face sizes. It noticeable that roughly two fifths of the tracks have quite low resolution. This influences the accuracy of face recognition.

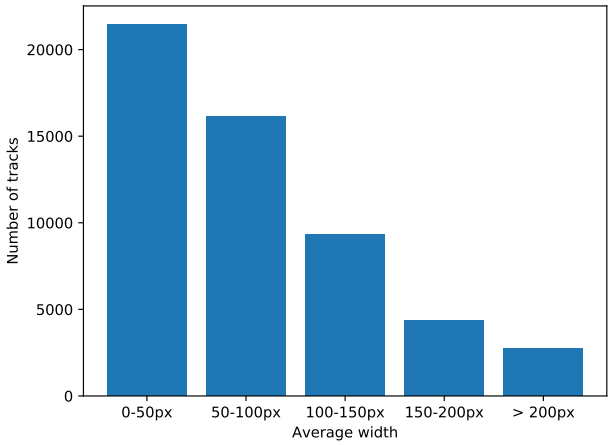


Figure 4.5: Distribution of face sizes computed for each track in manually annotated database.

It is important to note that just selecting trailers to have even distribution of age and gender over the starring celebrities might not be enough to get the even distributions over the annotated tracks. This is because some of the trailers might not even capture the celebrity that is associated with it. In addition, some trailers may contain higher number of tracks with the celebrity than others. Figure 4.6 shows how many tracks we have annotated for each age and gender categories. It can be seen that almost all age groups are represented in our database. The total number of tracks with actresses is $3,402$ and $3,217$ tracks contain actors hence the gender distribution of tracks is quite balanced.

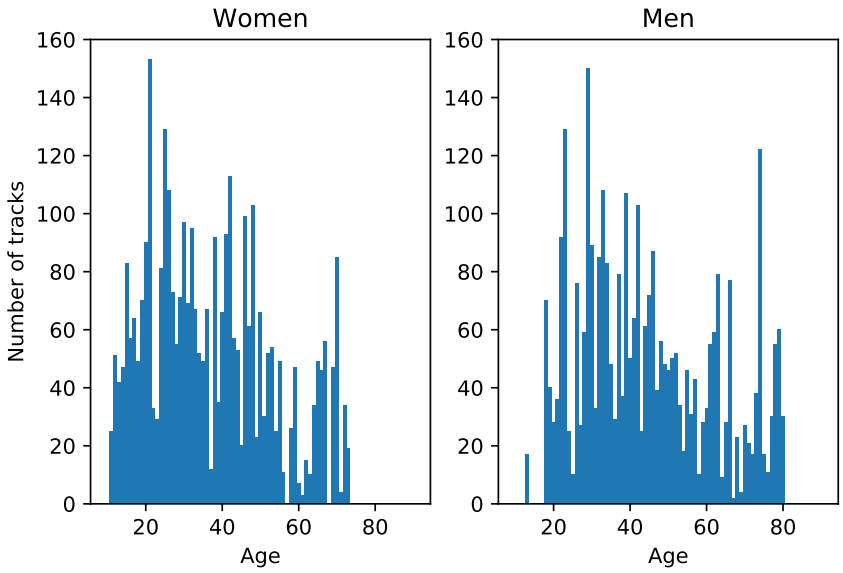


Figure 4.6: Histogram of the number of tracks for given age and gender category in the manually annotated database.

Length of the found tracks is highly variable. Some of the tracks contain only 5 faces. Some tracks are quite long and consist of hundreds of images. Figure 4.7 shows distribution of track lengths in our database only for tracks with less than or equal to 100 images which covers most of the cases. In particular, around 97.2 % of tracks contain 100 or less images. The average number of images per track is 27.6.

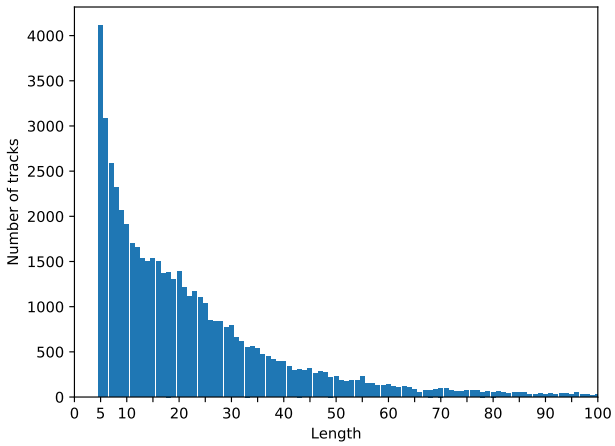


Figure 4.7: A histogram of the length of tracks in the manually annotated database.

Figure 4.8 shows the number of facial images in the manually annotated database for each gender and age category. That is, images from all annotated tracks put to one bag. The total number of annotated images is 231,608 where 116,643 of them capture females and 114,965 capture males, i.e. the gender distribution is balanced.

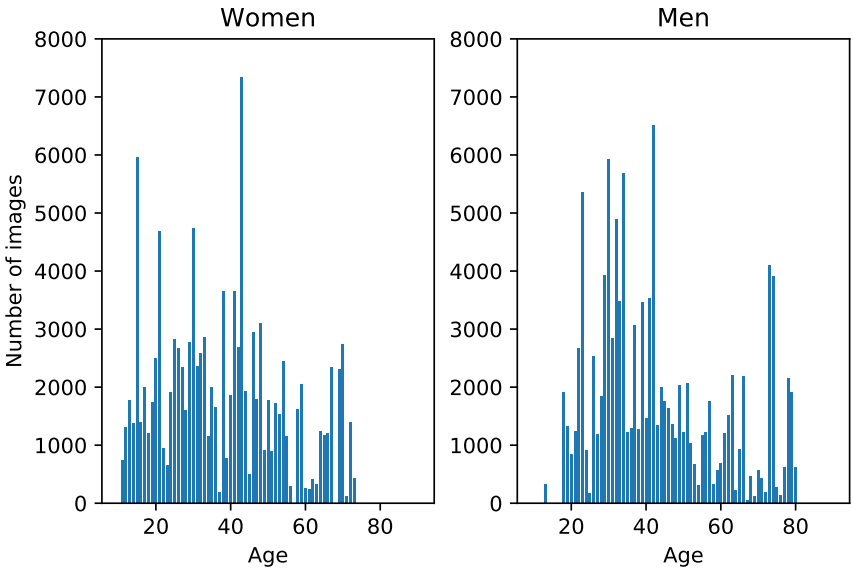


Figure 4.8: Histogram showing the number of facial images for each age and gender category in the manually annotated database.

Figure 4.9 shows how many TMDB portraits are available for celebrities selected for our manually annotated database.

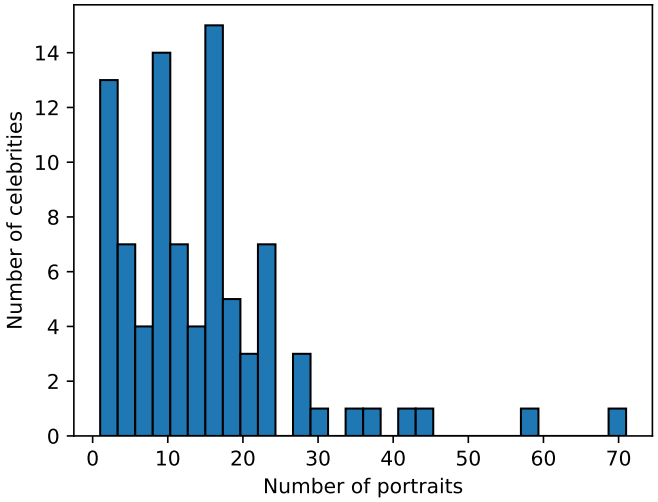


Figure 4.9: A histogram showing the number of portraits for celebrities contained in the manually annotated database.

The number of IMDB images available for each celebrity is highly variable. Popular celebrities have hundreds of images which is beneficial when we want to create their appearance models. On the other hand, less known celebrities usually have only a few images. We show experimentally that the number of pictures used to create an appearance model has a large impact on the face recognizer which is in the heart of the proposed annotation algorithm. A histogram of the number of IMDB images per celebrity is shown in Figure 4.10.

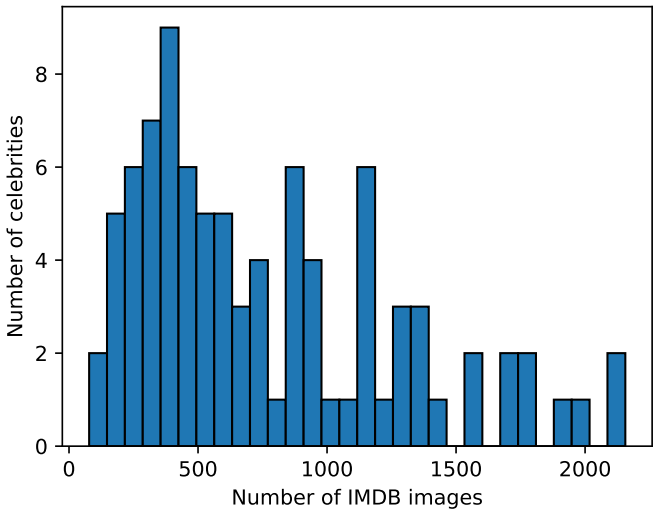


Figure 4.10: Histogram of the number of IMDB images for celebrities selected to the manually annotated database.

4.4 Annotators

Because the number of tracks to be manually annotated was quite high, the workload was distributed over several people. Table 4.1 shows number of tracks annotated by each annotator. Because the main goal was to annotate each track at least once the overlap of manual annotations is very low. Currently, we do not exploit multiply annotated tracks.

Annotator	# of tracks annotated
Me	38 943
Brother #1	10 029
Brother #2	3 086
Brother #3	2 396
Supervisor	1 156

Table 4.1: The number of manually annotated tracks by each annotator.

Chapter 5

Experiments

The proposed method uses multiple parameters and depends on multiple design options which influence its accuracy. These parameters and design options were more thoroughly described in Chapter 3. The following sections focus on tuning the method's parameters. Namely, Section 5.1 specifies a list of all tuned parameters, Section 5.2 analyses various bounding box multipliers and VGG-Face2 architectures, Section 5.3 describes used evaluation protocol, Section 5.4 focuses on tuning parameters when using portrait images. Section 5.5 tunes parameters of proposed annotation method when using IMDB images, Section 5.6 evaluates method's accuracy using tuned parameter on various categories of tracks, Section 5.7 provides a summary of best performing parameter configurations and Section 5.8 evaluates approaches that can be used to efficiently store a large number of face sequences.

5.1 Tuned hyper-parameters

The following list summarizes all parameters and design options we consider:

- Face bounding box multiplier which decides how much to extend bounding box found by face detector to get optimal face recognition performance
- VGG-Face2 architecture:
  - ResNet-50
  - SE-ResNet-50-256D
- Images used to create celebrity descriptors:
  - TMDB portraits
  - IMDB images
- Threshold θ used when creating celebrity descriptor to filter faces detected in IMDB images
- Method for aggregating face descriptors to a compact celebrity descriptor:
  - coordinate-wise median

- average

of identity feature vectors extracted from face images by VGG-Face2

- Approach used to aggregate distances between images in a track and celebrity descriptor into a single value:
  - median
  - average
- Threshold Θ used to decide which tracks belong to certain identity based on their distance to celebrity descriptor

5.2 Tuning face bounding box multiplier and VGG-Face2 architecture

Size of the face bounding box influences the accuracy of used VGG-Face2 face recognizer. The bounding box outputted by the face detector does not cover the whole head. By increasing its size the image will also include hair, chin and ears. The face recognition neural networks described earlier were pretrained on the MS-Celeb-1M [23] database and then fine tuned on the VGG-Face2 [24] dataset using face bounding boxes detected by MTCNN which were extended by a factor of 1.3 [21]. This means that both width and height were multiplied by 1.3 and the center of the bounding box stayed the same.

To evaluate which combination of bounding box multiplier and VGG-Face2 architecture works well on our manually annotated data we used the following method. From each annotated track we selected one face. To be specific, we selected image in the middle of each track. Because the tracks were manually annotated we know which celebrity is shown on these images. Then we created set of triplets $\mathcal{X} = \{(A_1, B_1, C_1), \ldots, (A_n, B_n, C_n)\}$ containing faces A, B and C. Faces A and B are different images of the same identity and face C is an image of different person. Then we use the following formula to measure accuracy for a particular parameter configuration:

$$score(A, B, C) = D(\phi(A), \phi(B)) - \min(D(\phi(A), \phi(C)), D(\phi(B), \phi(C)))$$

$$error(A, B, C) = \begin{cases} 0 & \text{if } score(A, B, C) < 0 \\ 1 & \text{else} \end{cases}$$

$$mistakes = \sum_{i=1}^{n} error(A_i, B_i, C_i)$$

where $D(\phi(A), \phi(B))$ is a cosine distance between feature vectors $\phi(A)$ and $\phi(B)$ extracted from face images A and B by VGG-Face2.

This method calculates the number of incorrectly ranked triplets, that is, in how many triplets the two feature vectors of different identities are more similar to each other than the two feature vectors of the same identity. By

using this approach we can evaluate performance of the used face recognition setting without the need to select a specific threshold on cosine distance.

Figure 5.1 shows the number of mistakes for two VGG-Face2 architectures and a range of face bounding box multipliers. It can be seen that the VGG-Face2 architecture **SE-ResNet-50-256D** with **bounding box multiplier 1.45** makes the least amount of mistakes thus it is the configuration we will use in further experiments. Specifically, the face bounding box multiplier and VGG-Face2 architecture parameters are used in the process of creating celebrity descriptors and in the the process of identifying celebrities in tracks. It does not make much sense to tune different combinations of aforementioned parameters for each of these processes. Therefore we will use the combination which proved to work best in this section for both building blocks of the whole algorithm.

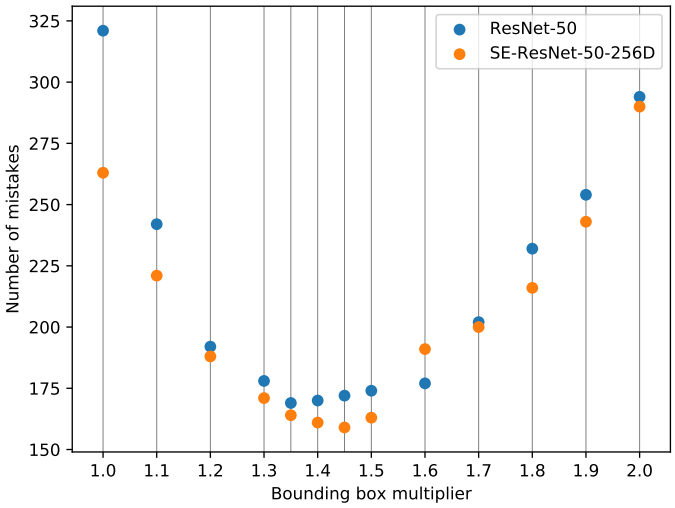


Figure 5.1: Accuracy of VGG-Face2 models measured on 2180 triplets of faces.

Figure 5.2 shows an example of various bounding box multipliers. It can be seen that without extending the bounding box the face image does not contain chin and ears. Additionally, the bounding box extended by a multiplier of 2.0 contains actor's surroundings which negatively influences the face recognition accuracy.

(a) : Multiplier 1.0 **(b) :** Multiplier 1.45 **(c) :** Multiplier 2.0

Figure 5.2: An example of face bounding box extended by various multipliers.

5.3 Evaluation protocol

The purpose of this section is to familiarize the reader with metrics and visualizations used to evaluate the annotation algorithm for various parameter settings. Figure 5.3 shows the percentage of incorrectly annotated tracks for two parameter configurations:

1. IMDB images median - Celebrity descriptors were created from all faces detected in images found on celebrity's IMDB page. The descriptor was formed by calculating coordinate-wise median of extracted identity feature vectors and the distances between celebrity descriptor and images in track were aggregated by using median.

2. Portraits median - The only difference in this configuration is that the celebrity descriptors were created from TMDB portraits (images containing only selected celebrity) instead of IMDB images. The other parameters were the same.

The x-axis represents the threshold Θ used for deciding whether track is sufficiently close to the celebrity descriptor and thus can be selected as belonging to the celebrity. It can be seen that the configuration using IMDB images works reasonably well, considering its simplicity, although the celebrity descriptors created from portraits perform noticeably better.

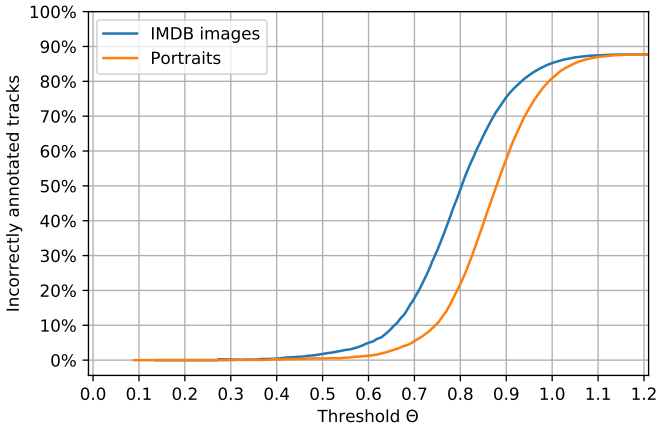


Figure 5.3: Percentage of incorrectly annotated tracks of two baseline parameter configurations.

To fully evaluate each configuration we also have to take into account the percentage of extracted tracks. Imagine that, for instance, we are able to reach low percentage of incorrectly annotated tracks. However, this metric does not tell what portion of all tracks belonging to the target celebrities are actually found by the algorithm. The manually annotated database contains 6,619 tracks with known identities. For example, if we are able to correctly find 3,000 of them by the algorithm then the percentage of extracted tracks is 45%.

Figure 5.4 shows the percentage of extracted tracks for the configurations defined above. It is observable that the configuration using TMDB portraits yields lower percentage of extracted tracks for threshold $\Theta > 0.34$. However, Figure 5.3 shows that the percentage of incorrectly annotated tracks is much lower in comparison to the other configuration.

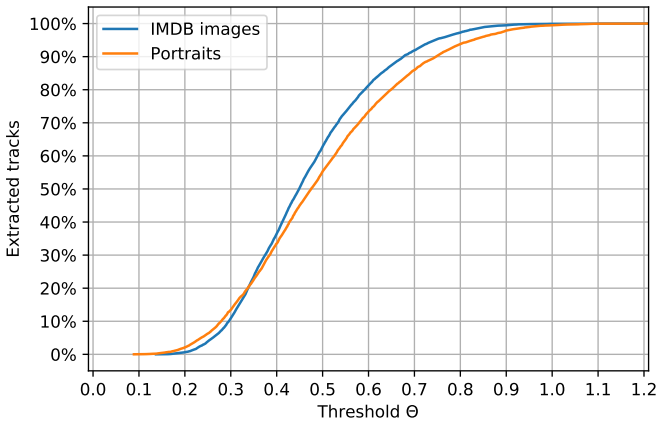


Figure 5.4: Percentage of extracted tracks for two baseline parameter configurations.

It is not immediately clear from the two figures above which configuration performs better. We can plot both of the metrics into one chart showing

their relation for the full range of threshold Θ. The only disadvantage is that it isn't possible to also show the threshold on the chart without the chart becoming too complicated. However, the threshold values are not that important at this stage where the goal is to tune other parameters first.

Figure 5.5 shows the aforementioned relation of the two used metrics. Note the changed limits of y-axis. Now it is clear which configuration performs better. It is the one with lower percentage of incorrectly annotated tracks for the same extraction percentage.

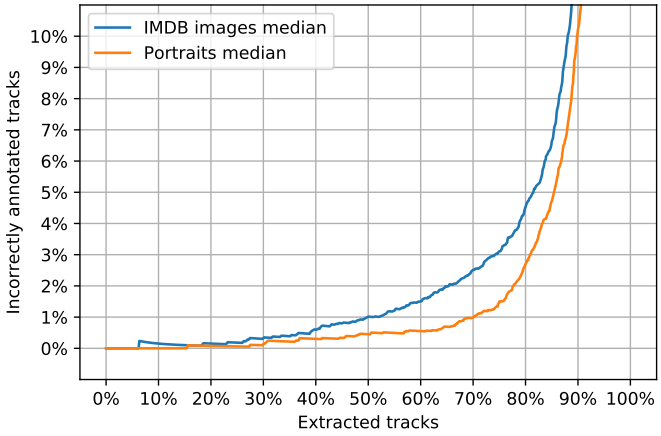


Figure 5.5: Relation between the percentage of incorrectly annotated tracks and percentage of extracted tracks.

The following two sections focus on tuning parameters separately for both sources of celebrity images.

5.4 Parameter tuning when using portrait images

Figure 5.6 shows the difference in annotation accuracy when aggregating extracted identity feature vectors by either calculating average or coordinate-wise median. It can be seen that the value of this parameter plays almost no role in the currently used setting. For the further experiments we will use coordinate-wise median for aggregating identity feature vectors into celebrity descriptors.

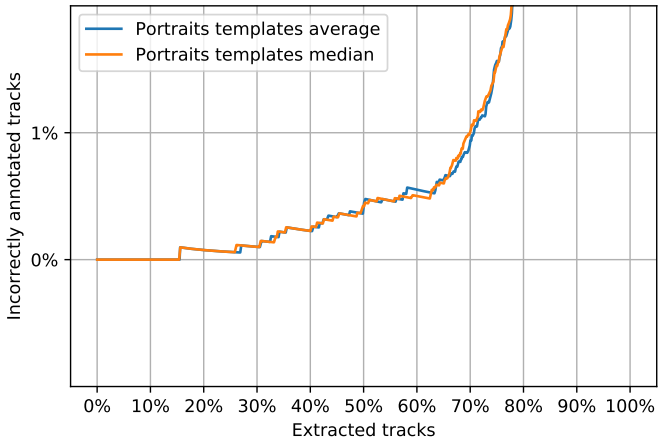


Figure 5.6: Accuracy of two different approaches to computing celebrity descriptors.

Figure 5.7 shows how aggregating distances between celebrity descriptors and track images influences annotation accuracy. We evaluated two options. One calculates average and the other median of said distances. It can be seen that the difference in method's performance for both options is very small. However, using average distance perform slightly better thus will be used for further experiments.

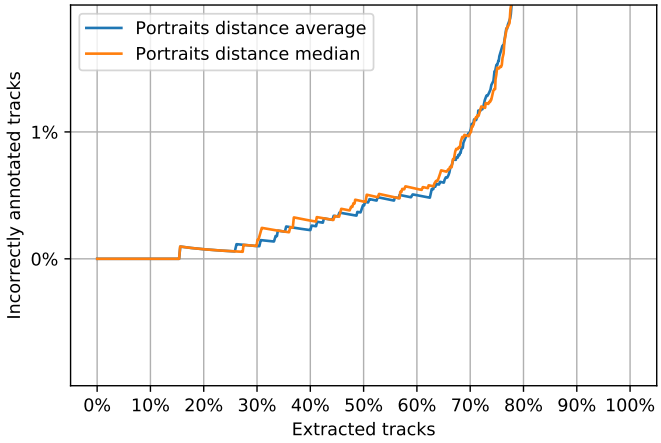


Figure 5.7: Annotation error of two different approaches to calculating the distance between a celebrity descriptor and a face track.

5.4.1 Annotation error versus number of portraits

The evaluations above don't take into account the number of portraits used to create celebrity descriptors. The number of portraits available is somewhat limited. Therefore we need to estimate how many portraits are required to create sufficient identity representation.

Figure 5.8 compares the performance of celebrity descriptors created from 10 portraits and less. Not all of the celebrities selected for the manually

annotated database have 10 portraits available on TMDB website, hence the following figure is measured on a subset of 34,817 tracks. It can be seen that the configuration using 10 portraits performs slightly better than the one using only 5 portraits for creating celebrity descriptor.

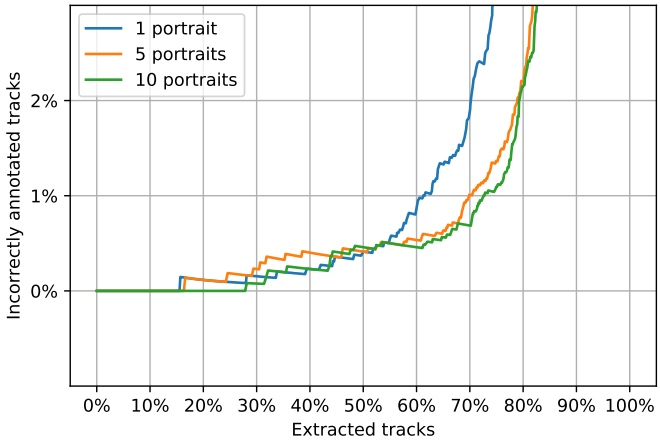


Figure 5.8: Annotation error when using celebrity descriptors created from a subset of portraits.

Figure 6.1 shows that the number of celebrities with 10 and more portraits is very low. Figure 5.9 compares celebrity descriptors created from 1, 2, 3 and 5 portraits. This evaluation was performed on subset of 46,724 tracks because we don't have 5 portraits available for some celebrities. It can be seen that up to a point all configurations perform similarly. For percentage of incorrectly annotated tracks greater than 0.5% there is a noticeable difference and more portraits means less errors for the same extraction percentage.

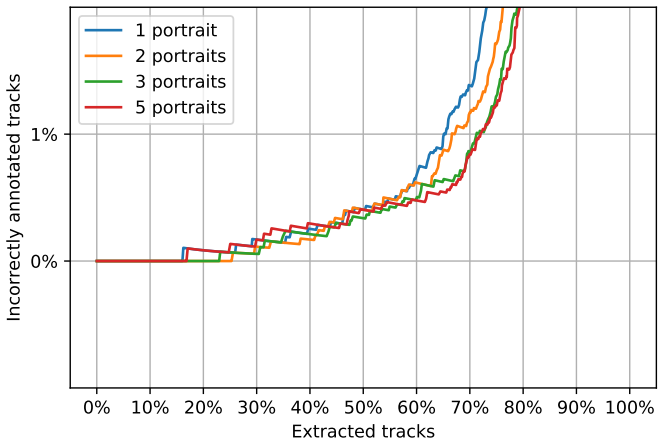


Figure 5.9: Annotation error when using celebrity descriptors created from different subset of portraits.

5.5 Parameter tuning when using IMDB images

The method used for filtering images belonging to selected celebrity was introduced in Section 3.5.2. This method requires threshold θ as a parameter. Purpose of this threshold is to select only a subset of faces that likely contain selected celebrity.

Figure 5.10 shows annotation error when creating celebrity descriptors from all face images detected in celebrity's images. The second configuration uses only the face which is closest to the coordinate-wise median (created from identity feature vectors of all detected faces) from each image. As expected, this partial filtering considerably lowers the annotation error.

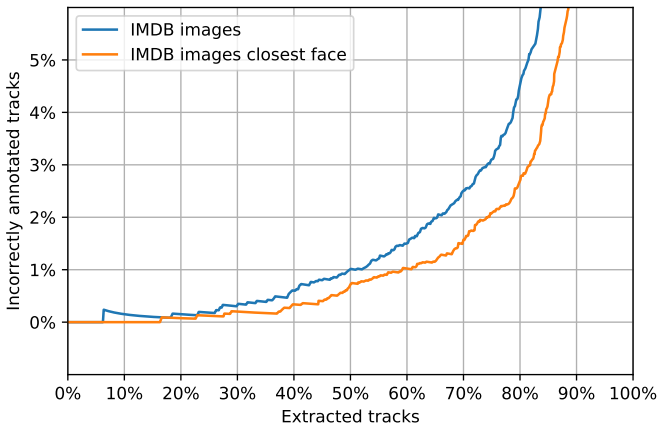


Figure 5.10: Annotation error of celebrity descriptor built from partially filtered faces.

5.5.1 Finding optimal filtering threshold

To evaluate which filtering threshold θ works best we can choose maximum allowed percentage of incorrectly annotated tracks and compare the percentage of extracted tracks for a range of thresholds. Because the percentage of incorrectly annotated tracks is not continuous we choose the greatest extraction percentage that leads to annotation error lower than the set value.

For example, if we say that we want to create database with at most 0.5% tracks incorrectly annotated we will get the extraction percentages shown in Figure 5.11. For this figure we evaluated thresholds θ from range 0.34 to 1.0 with step of 0.02. It is observable that using threshold $\theta = 0.40$ we can extract 59% of tracks and still keep the percentage of incorrectly annotated tracks below 0.5%.

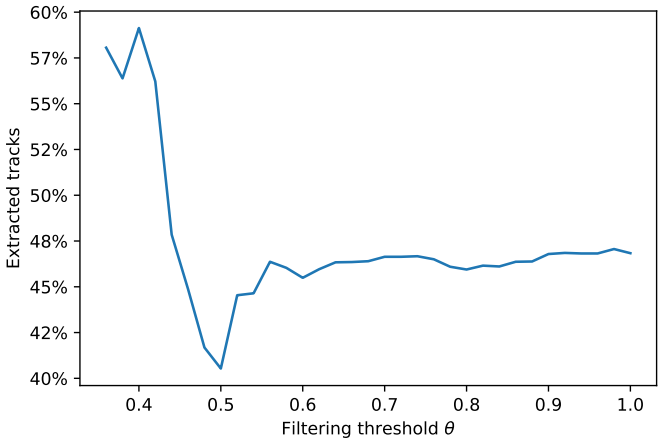


Figure 5.11: Percentage of extracted tracks for a range of thresholds θ and set maximum annotation error.

The lower bound of the threshold θ was selected so we could evaluate all thresholds θ on the same data (whole set of tracks). The lowest distance between closest face and the median (the one used for filtering) is different for each celebrity. For some it goes as low as 0.10. But for most it is higher. By using $\theta \geq 0.34$ it is guaranteed that we are able to create descriptor for every celebrity. This, of course, holds true only for the celebrities in the manually annotated database.

Figure 5.12 shows performance of annotation method using celebrity descriptors built from IMDB images filtered with threshold $\theta = 0.40$. The difference in accuracy is easily noticeable from the figure.

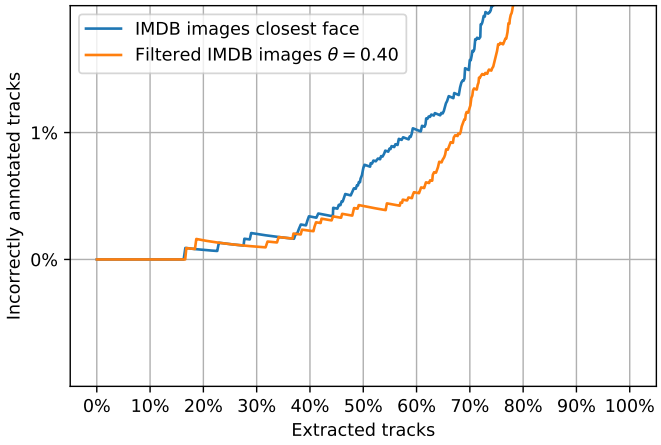


Figure 5.12: Annotation error when using celebrity descriptors built from faces filtered by threshold θ.

5.5.2 Celebrity descriptor aggregation

Figure 5.13 shows the difference in annotation error when aggregating extracted identity descriptors by either calculating average or coordinate-wise

median. It can be seen that the latter performs slightly better. Therefore we will use this parameter configuration for further experiments.

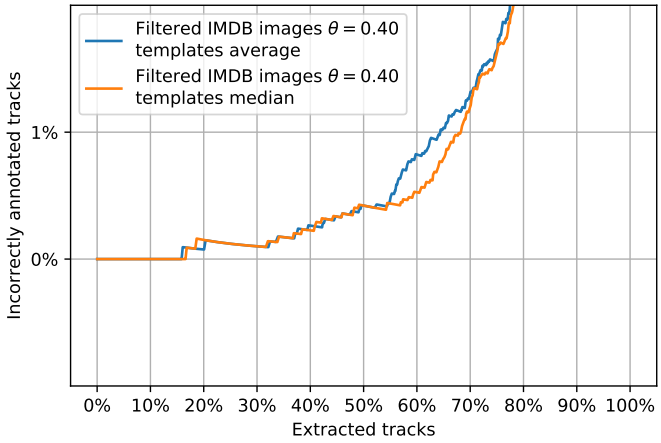


Figure 5.13: Comparison of different techniques used to create celebrity descriptors.

5.5.3 Celebrity descriptor to track distances

Figure 5.14 shows the change of performance when aggregating distances using both of the aforementioned options. It can be seen that using average distance performs slightly better hence will be used for further experiments.

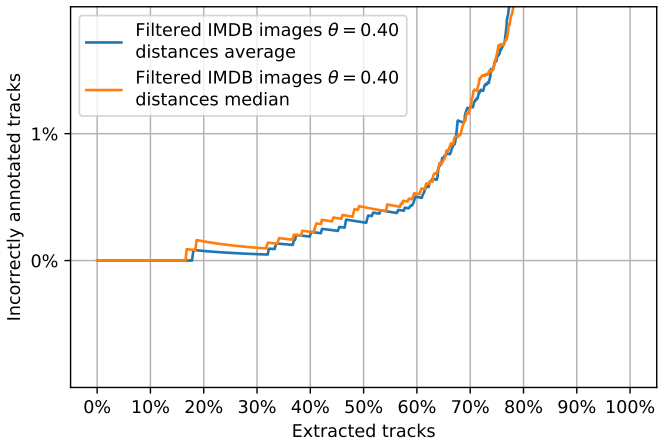


Figure 5.14: Comparison of different techniques used to aggregate distances of each image in track to the celebrity descriptor.

5.5.4 Number of images versus annotation error

The number of images used for creating each celebrity descriptor influences the accuracy of the annotation method. Generally, the number of IMDB images available to us is much larger than the number of portraits. This allows us to use more celebrity's faces to create their descriptors. The number of

IMDB images for celebrities in the manually annotated database is shown in Figure 4.10. This distribution shows that big portion of the selected celebrities has high number of images. However, this is caused by the selection criteria that we used for choosing these celebrities. We chose popular celebrities and popular celebrities have more images available than average celebrities.

We can measure how well does the annotation method perform if we randomly select a subset of celebrity's images and then apply the previously tuned filtering approach. Figure 5.15 shows method's annotation error when randomly selecting 25, 50, 75 and 100 images. Because some of the selected celebrities do not have 100 images the measurement is performed on a subset of 50,144 tracks. It can be seen that using 50 and more IMDB images yields similar results.

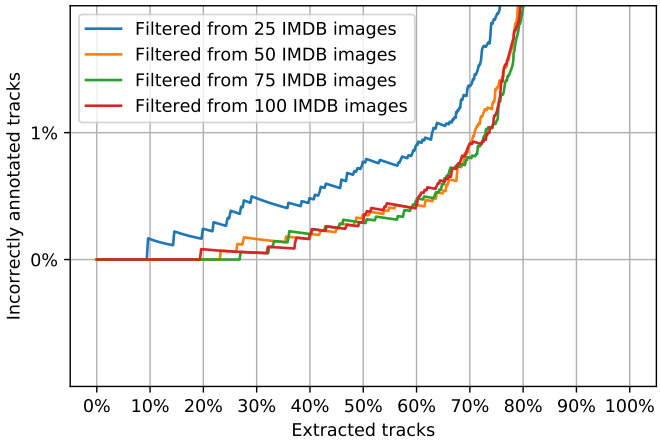


Figure 5.15: Annotation error when using celebrity descriptors built from a subset of images.

5.6 Evaluation on different track categories

We can evaluate the configuration tuned for each source of images described above on various categories of tracks. Tracks can be split into multiple categories by following attributes:

- **By age** - We only know the age for tracks that were manually annotated as containing the target celebrity. Because of this, we put all tracks (even if they show unknown identity) found in videos where the age of the celebrity we are looking for belongs to a certain range. By doing it this way we can evaluate the amount of incorrectly annotated tracks.
- **By gender** - Tracks were split similarly as for the age category but based on celebrity's gender.
- **By face size** - Because each track contains a sequence of faces we use an average size. The size of face is specified by width and height. We could also describe it by area as a single number but that makes it harder to

visualize how big the images actually are. We used the average width which is enough to describe the size of a face in our case. The face bounding boxes do not have a set width to height ratio but it does not vary that much. Using only width to categorize the data is easier to understand.

5.6.1 Age

In the testing database we have following number of tracks found in videos where the age of target celebrity belongs to following ranges:

- $\leq$ 18 years old - 3,420 tracks, 14.8% of them belong to the target celebrity,
- 19 to 45 years old - 26,811 tracks, 13.7% of them belong to the target celebrity,
- $>$ 45 years old - 23,763 tracks, 10.3% of them belong to the target celebrity.

These ranges were chosen to somewhat represent young, middle aged and older celebrities. For example it may be harder to identify young people because their appearance is not as stable as for middle aged people.

Figure 5.16 shows that younger celebrities are harder to identify on tracks which was expected.

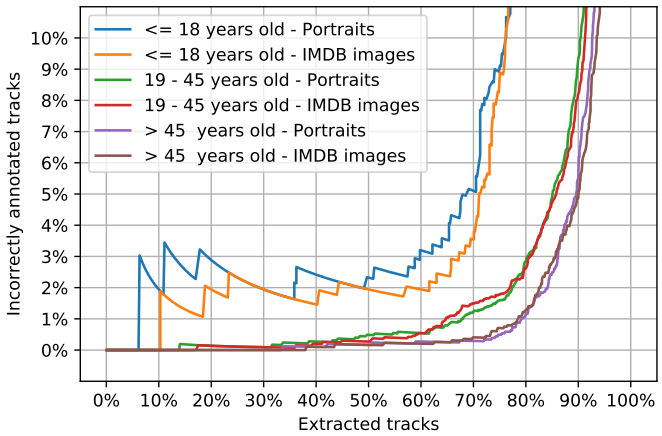


Figure 5.16: Evaluation on tracks split into multiple age groups.

5.6.2 Gender

The appearance of actors and actresses is usually different in real life and while playing a movie character. Also the appearance of actresses is not as stable as appearance of actors. It is more likely for women to change her hairstyle or hair color during their career. These assumptions may affect the accuracy of created celebrity descriptors. We have the following number of tracks for both genders:

- Women - 27,075 tracks, 12.6% of them belong to the target actress,
- Men - 26,919 tracks, 12% of them belong to the target actor.

Figure 5.17 evaluates performance using the tuned parameters for men and women. It is clear that identifying women is harder.

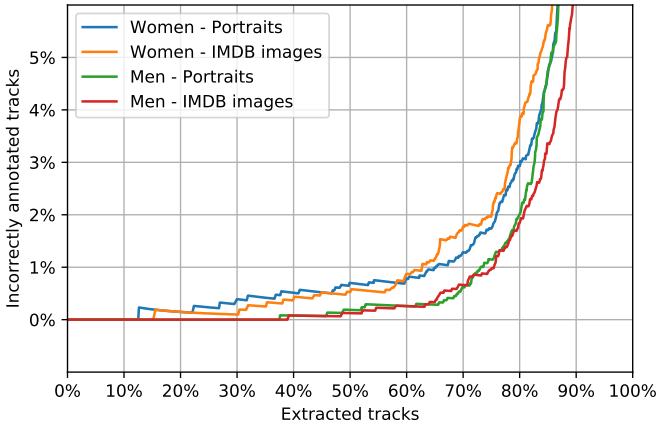


Figure 5.17: Evaluation on tracks split by gender.

5.6.3 Face size

The neural network we use for extracting identity feature vectors requires images of size 224×224 pixels. This means that we have to resize each face to this shape. This may be a problem for faces smaller than this shape because the face has to be stretched out and ends up blurry. Feeding the neural network with those low quality images may lead to weak performance.

The following figure shows how image quality influences the annotation error of proposed method. We split the annotated tracks based on average width of face in track into following categories:

- $<$ 50px - 21,449 tracks, 6% of them belong to the target celebrity,
- 50 - 149px - 25,438 tracks, 15.6% of them belong to the target celebrity,
- $\geq$ 150px - 7,107 tracks, 19% of them belong to the target celebrity.

As you can see from the numbers above the percentage of tracks annotated with celebrity is much lower for tracks with width lesser than 50 pixels. This is probably because the roles that are significant for the movie get more exposure. Usually the important characters take greater part of the screen compared to supporting roles. You can imagine a movie shot of stadium audience which contains many faces but the probability that many of them are popular celebrities is very low.

Figure 5.18 shows annotation error on tracks divided by their size. It can be seen that the bigger the face is the lower percentage of incorrectly annotated tracks is. It is also noticeable that most of the mistakes are made on low quality face images.

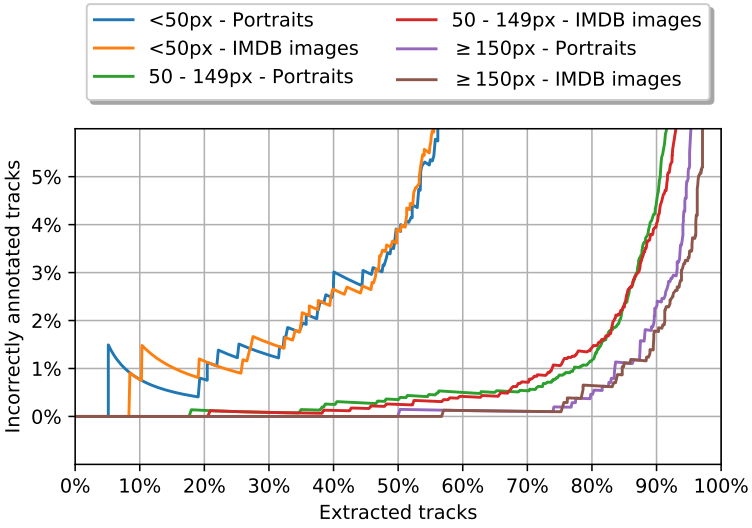


Figure 5.18: Evaluation on tracks divided into groups by their size.

5.7 Summary

We found the following configuration to be the most accurate for both sources of images:

- Bounding box size multiplier: 1.45
- VGG-Face2 architecture: SE-ResNet-50-256D
- Method for aggregating identity feature vectors to a compact celebrity descriptor: coordinate-wise median
- Approach used to aggregate distances between images in a track and celebrity descriptor into a single value: average
- Filtering threshold for IMDB images: $\theta = 0.40$

As mentioned, this parameter and design options configuration works best for both sources of images. The only difference is that we use threshold θ to filter IMDB images whereas TMDB portraits do not have to be filtered.

It cannot be simply said which source of images is better. As we have shown earlier, the more images we use the better (up to a point). Figure 5.19 compares annotation error when using 5 portraits, 25 (25 images before filtering faces) and 50 IMDB images (50 images before filtering faces). It can be seen that using 50 IMDB images to filter faces and create celebrity descriptors leads to lower annotation error compared to 5 portraits. On the other hand, using only 25 IMDB images leads to higher annotation error.

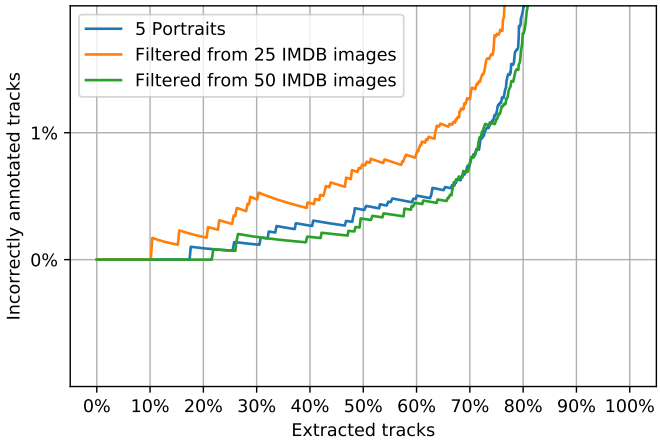


Figure 5.19: Evaluation of the tuned method on both sources of celebrity images.

For the whole process of automatic track annotation both image sources can be used to complement each other. For clarify, lets say we have two subsets of celebrities. First subset contains celebrities with 5 and more TMDB portraits. Second subset contains celebrities with 50 and more IMDB images. We found out that the overlap of those subsets is very low. This observation can be used to cover higher number of celebrities when creating large database. Table 5.1 shows how many celebrities fulfil multiple conditions of minimum number of images. It can be seen that using by 50 IMDB images or 5 TMDB portraits we are able to create sufficient celebrity descriptor for larger number of celebrities. As show in Figure 5.9 we can use even less portraits than 5 to achieve similar annotation accuracy. Figure 6.1 shows that the number of celebrities with, for instance, at least 3 portraits is substantially higher.

Condition	#Celebrities
$\geq$ 5 portraits	1,291
$\geq$ 50 IMDB images	8,351
$\geq$ 5 portraits or $\geq$ 50 IMDB images	9,002
$\geq$ 5 portraits and $\geq$ 50 IMDB images	640

Table 5.1: Number of celebrities for various selection criteria.

We did not set the value of threshold Θ. This threshold is used to decide which tracks belong to certain celebrity based on their distance to their descriptor. It cannot be said which value is best. It is a trade-off between annotation error and the number of extracted tracks. Low threshold leads to lower number of annotation errors but also leads to a lower number of extracted tracks. Figure 5.20 shows that up to accuracy error of 0.5% both configurations perform similarly. Imagine that, for instance, you want to create automatically annotated database that is very accurate. In that case, you can simply choose threshold $\Theta = 0.30$. As Figure 5.21 shows, the amount of extracted tracks will be low which is the trade-off for more accurate annotations.

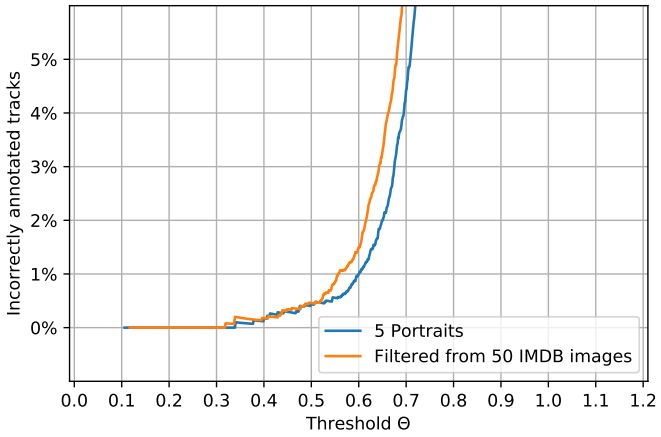


Figure 5.20: Annotation error for a range of thresholds.

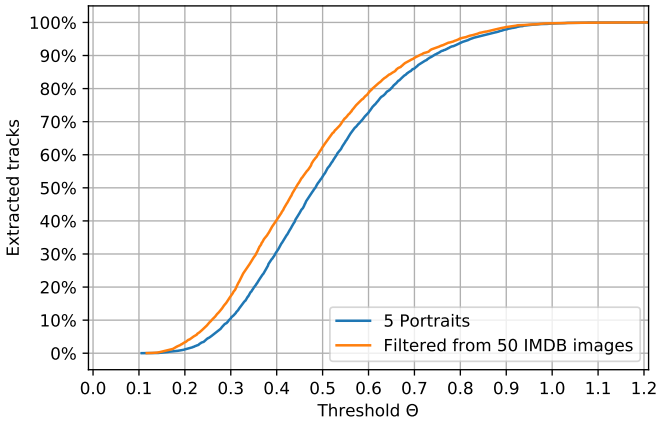


Figure 5.21: Percentage of extracted tracks for a range of thresholds.

5.8 Track compression

The created database of automatically annotated tracks will be quite large in terms of disk storage required to save it. We can use data compression to decrease its size. Considering that we are creating database of image sequences we can use video encoder to encode images from each track into video. The changes between each frame in track are minimal and video encoders can use this to their advantage. By storing the sequence of images in track as video we can save some space even if we use lossless compression. Lossless compression means that after decompressing the compressed data we do not lose any information. Lossless compression might not be that useful in our case because the source videos downloaded from the Internet are already compressed.

We can also compress the data and lose some information. By doing that, we can reduce the disk size required even more. We will try to experimentally show how much information we can lose to keep the database useful. Simply

said, if we compress the data too much the database might become useless because the outputted video will be of low quality. There are multiple video encoders that we can use for compression. This section compares two of those, specifically H.264 [25] and H.265 [26]. When compressing video with aforementioned encoders we can define constant rate factor(CRF). The range of CRF scale is 0-51 where 51 is the worst quality possible and 0 is a lossless compression. As [25] mentions, CRF 18 is subjectively lossless and it looks nearly the same for the human eye but it isn't technically lossless. To somehow chose the best CRF for our use case we can compare the accuracy of used face recognition approach using various constant rate factors.

We can compare the accuracy of face recognition evaluated on original images and on compressed images. If the accuracy is similar then we can use the CRF used to compress these images. The following figure shows the number of mistakes made on 2,180 triplets using the same method introduced in Section 5.2. Figure 5.22 shows that even when using $CRF = 20$ compression rate we can achieve similar results.

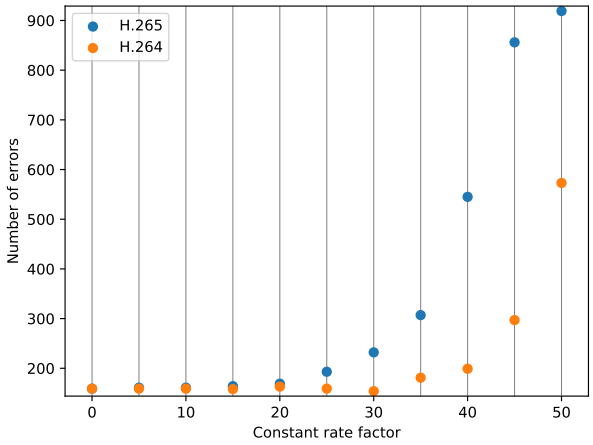


Figure 5.22: Number of mistakes made on compressed images from the manually annotated database for various constant rate factors.

To evaluate the used encoder more thoroughly we also have to consider its other aspects. Figure 5.23 shows how much disk space we can save with different constant rate factors. The results shown in the figure were measured on annotated tracks found in a subset of 100 videos from the automatically annotated database described in Chapter 6. The original size of annotated tracks when using threshold $\Theta = 0.5$ without any compression was 2,417 MB.

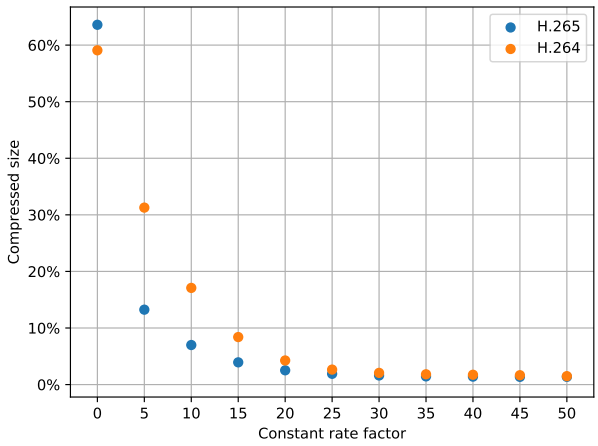


Figure 5.23: Comparison of database sizes for multiple constant rate factors.

We also have to take into consideration how much time it takes to compress the data. Figure 5.24 shows comparison between the two encoders used for compression. It can be seen in the figure that the encoder H.264 is much faster compared to encoder H.265. The trade-off between H.264 and H.265 encoders is saved disk space for CPU time spent on compression. For $CRF = 20$, the H.264 encoder performs almost identically to the H.265 algorithm. The disk space to store the database is almost identical with respect to the original size. Relatively to size of the compressed data, the size of the database when using H.265 is two times lower than when using H.264. On the other hand, using H.264 is more than 3 times faster than H.265 encoder.

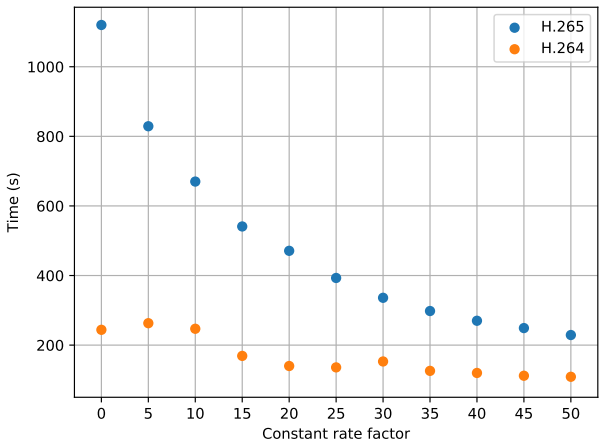


Figure 5.24: CPU time spent on compressing the database in seconds.

Chapter 6

IMDB video faces

We used the information gained from evaluating the proposed annotation algorithm on the manually annotated database to identify the optimal parameter setting of the method. The algorithm with the found parameter setting was used to create a large database of video tracks annotated with age, gender and identity. We named the created and automatically annotated database the IMDB video faces. This chapter describes details of the process, namely, Section 6.1 summarizes the used parameters of the annotation algorithm, Section 6.2 describes how the celebrities were selected and Section 6.3 which trailers were selected to the database. Distribution of tracks obtained by processing the selected trailers with a face tracker is described in Section 6.4. Section 6.5 reports analysis of results provided by the proposed annotation algorithm applied on the tracks. Finally, Section 6.6 summarizes main statistics of the IMDB video faces database and compares it with the Accio database.

We used the following sources of input data for the proposed annotation method:

- IMDB datasets (available on their website):
  - title.basics dataset which provides information about movies like release year.
  - name.basics dataset which contains data about celebrities such as birth year, name or their IMDB identifier.
- Videos:
  - downloaded from celebrity's video gallery on IMDB website. For example `www.imdb.com/name/nm0105672/videogallery/` where *nm0105672* is the IMDB ID of celebrity that can be found in the dataset mentioned above.
- Images:
  - IMDB images are downloaded from celebrity's image gallery. For instance, images of Andre Braugher are listed on the following webpage `www.imdb.com/name/nm0105672/mediaindex/` where *nm0105672* is his IMDB identifier.

  - TMDB portaits were downloaded using available API described below. They are also available on their website, for instance pictures of Morgan Freeman can be found here `https://www.themoviedb.org/person/192-morgan-freeman/images/profiles`

- The Movie Database API:
  - method which for given celebrity's IMDB ID returns their TMDB ID if it exists. Further description can be found in the API documentation `https://developers.themoviedb.org/3/find/find-by-id`
  - method providing images for person defined by their TMDB ID described on the following website `https://developers.themoviedb.org/3/people/get-person-images/`

6.1 Parameter setting

For creating this database we used the following setting:

- Face detector: MTCNN
- Face bounding box multiplier: 1.45 [1]
- VGGFace2 architecture for extracting face feature vectors: SE-ResNet-50-256D
- Source of images: TMDB portraits
- Minimum number of images for creating appearance models: 5
- Method used to aggregate face feature vectors into celebrity descriptors: coordinate-wise median
- Approach used to aggregate distances between images in a track and celebrity descriptor into a single value: median
- Video encoder[2]: H.264 [25]
- Constant rate factor for compression: 20

6.2 Celebrity selection

The celebrities were chosen from the IMDB dataset based on their primary profession. We filtered only actors and actresses and ended up with around 300,000 celebrities. Then we investigated how many portraits each celebrity in the database has available using the TMDB API. The celebrity selection

[1] Width and height of bounding boxes were 1.45 times larger than the bounding boxes provided by the face detector.

[2] Used to encode track images into videos.

for our database is based on the minimum number of portrait images the celebrity has.

In Section 5.4.1 we showed that accuracy of the annotation algorithm increases with the number of portraits. On the other hand, the number of celebrities with high number of portraits is small. Hence we had to select a reasonable trade-off between accuracy of the produced annotations and the number of celebrities included in the database. We set the minimum number of portraits to be 5 because the algorithm performs reasonably well and the amount of celebrities is still large, namely, the number of celebrities with at least 5 portraits is 1,276; 418 of them are males and 858 are females.

Figure 6.1 shows a histogram of the number of identities who have at least given number of portraits images. The number of identities with 1 and more portrait images was excluded from the figure for clarity because its value is 78,271 and the figure would be harder to read. The figure shows that there are more females than males for all settings of the minimum number of portraits. For 1 and more portraits it is actually the opposite. 44,157 of males have at least 1 portrait and there are 34,114 females that have 1 or more portraits.

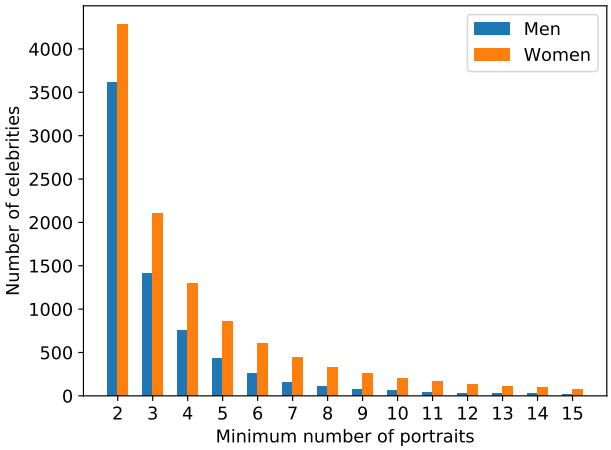


Figure 6.1: The figure shows the number of celebrities who have at least given number of portrait images.

6.3 Trailer selection

The idea behind selecting which trailers to process was the same as the one we used when creating the manually annotated database. Namely, we wanted the created database to have even distribution of age and gender. Second, we wanted to maximize the number of identities in the database to make it as diverse as possible.

Figure 6.2 shows how many trailers are available for different age categories. The figure takes into account only identities that fulfill the minimum number of portraits requirement, i.e. they have at least 5 portraits. The total number

of trailers for actresses is 20,106 and 15,445 for actors. It's noticeable that the women in available trailers are mostly within the age of 20 and 40 years old. For men, the peak is not as sharp and is around 40 years old.

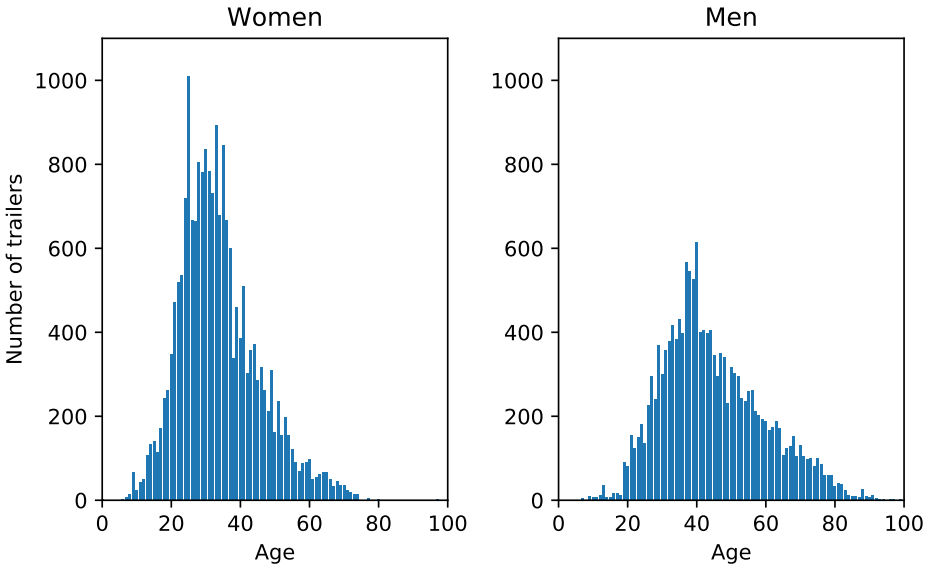


Figure 6.2: Number of trailers for age and gender categories.

The trailer selection was done using Algorithm 1. We limited the number of selected trailers for each identity to 10 which should help to keep the portion of tracks each celebrity has more evenly distributed. Limiting the number of selected videos for each gender to 2,000 was done because of the time it takes to process each video. We estimated the time required to process 4,000 videos to 38 days. The time estimate is based on processing 100 videos. Note

that this estimation considers using only one computer at a time.

```
Input  : list of trailers
Output : list of selected trailers
organize trailers into dictionary where key is gender and age and the
  value is list of trailers containing celebrities with corresponding age
  and gender;
foreach G ∈ [male, female] do
    while number of selected trailers for gender G < 2000 do
        foreach A ∈ [15, 80) do
            get trailers for specified age A and gender G and choose
              trailer that stars celebrity with lowest number of already
              selected trailers and remove it from the list;
            if number of selected trailers for celebrity starring in
              chosen trailer < 10 then
                mark trailer as selected;
            end
        end
    end
end
```

Algorithm 1: Algorithm used for selecting trailers.

Algorithm 1 selected 2,039 trailers containing women and 2,036 trailers containing men. Figure 6.3 shows the number of trailers selected for each age and gender category. It is observable that the number of trailers for actresses in age group from 60 to 80 years is slightly lower compared to the other age groups. That may be caused by the limit on number of selected trailers for each identity. Imagine that most of the trailers in aforementioned age group belong to a limited number of actresses. Even if there is enough trailers for this age group only a fraction of them can be used.

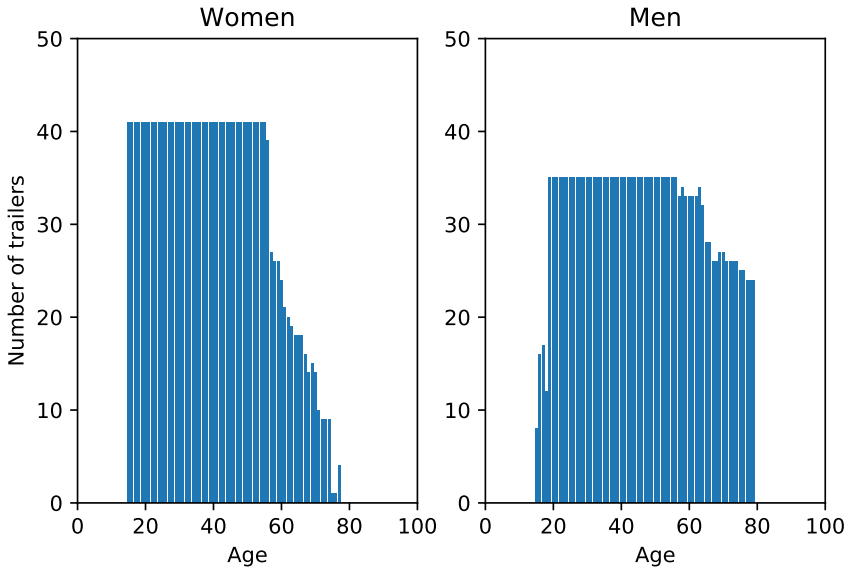


Figure 6.3: Number of selected trailers for age and gender categories.

Figure 6.4 shows how many trailers were selected for each celebrity. It is seen that the number of trailers for each identity is relatively well distributed due to the hard limit on maximum number of selected trailers we set. Also the average number of trailers selected for each actress is slightly lower than for actors because there were more actresses to choose trailers from. The total number of celebrities with selected trailers is 588 where 369 of them are actresses and 219 are actors.

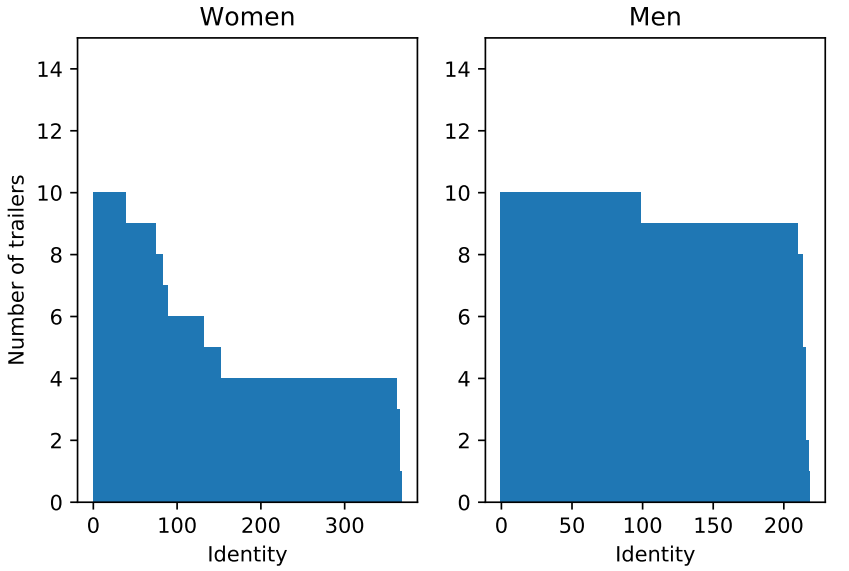


Figure 6.4: Number of selected trailers for each celebrity.

6.4 Tracks summary

Having the trailers selected, as described in the previous section, each trailer was processed by the face tracker. This section summarizes distribution of tracks that have been found.

- Total number of tracks: 391,454
- Average number of tracks per video: 96.06
- Total number of images: 11,841,096
- Average length of track: 30.25 images
- Longest track: 7,940 images, found in 4 minutes 26 seconds long interview without any cuts

6.5 Annotating tracks

To automatically annotate the tracks found in the selected videos we set the distance threshold between identity template and track to 0.512. Tracks with distance lower than this threshold are associated with the celebrity whose identity template was used. The threshold selection is based on the analysis described in Chapter 5. In particular, with the distance threshold set to 0.512 we are able to extract more than 50% of tracks associated with the selected celebrities while keeping the percentage of incorrectly annotated tracks below 1%. The percentage of extracted tracks is not that important once it reaches reasonable value because the amount of trailers available is very large and we can add more videos to process and create larger database.

The main objective was to keep the percentage of incorrectly annotated tracks low. Because great amount of mistakes were measured on small faces we take into account only tracks with average width of the contained faces greater or equal to 100 pixels. With this choice, the estimated percentage of incorrectly annotated tracks in the created database is 0.5%. This estimation is based on the database manually annotated by humans.

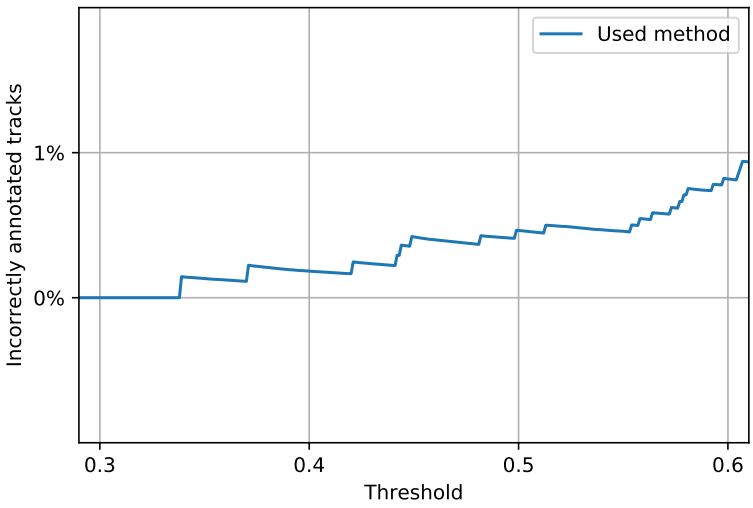


Figure 6.5: Estimated annotation error of created database.

Now that the approach for automatic track annotation is set we can further analyze the created database. As mentioned earlier, selecting trailers evenly with regard to age of celebrity starring in them does not mean the annotated tracks will also be evenly distributed.

The final number of tracks capturing the celebrities is 14,457. 7,353 of them belong to actresses and 7,104 to actors, hence the gender distribution is even as planned. Figure 6.6 shows the distribution of tracks with respect to age category. Although the age distribution of selected trailers was close to uniform, the age distribution of tracks is noticeably different. It can be seen that the track distribution is still reasonably balanced although it would be better to have more tracks for older women.

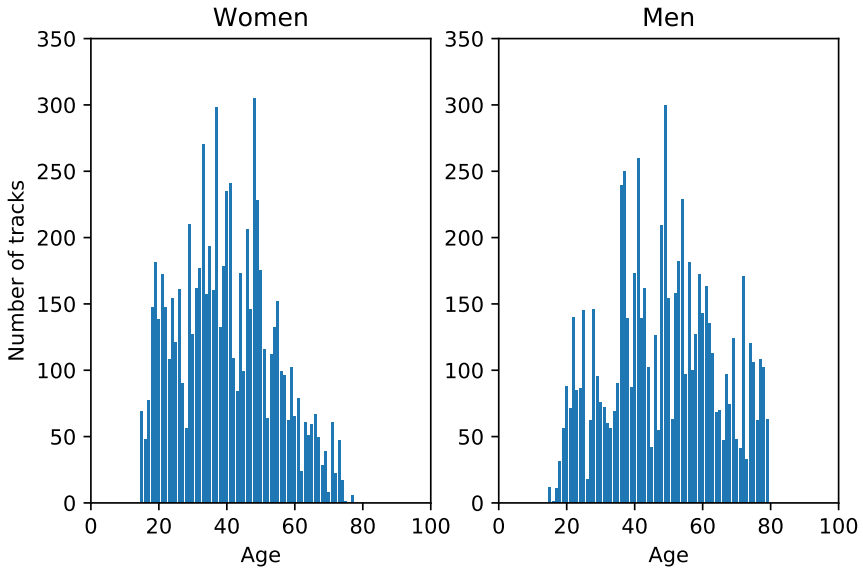


Figure 6.6: Distribution of annotated tracks for different age and gender categories.

Figure 6.7 shows distribution of tracks for individual celebrities. The celebrity with highest number of tracks is Wilson Cleveland. The created database contains 216 tracks of him which is around 1.5%. The average number of tracks per celebrity is 26 and the average number of face images per celebrity is 1,278.

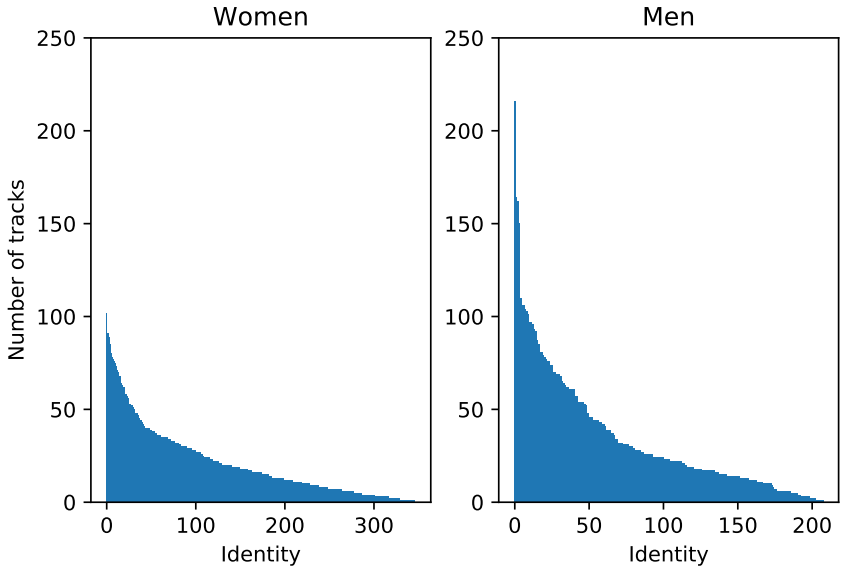


Figure 6.7: Number of annotated tracks for each celebrity.

6.6 Summary

Table 6.1 shows the main statistics of the automatically annotated database along with a comparison to the Accio database being the most relevant existing database we are aware of. Compared to the Accio database, the created database offers more diverse set of tracks in terms of the number of identities and different movies they were captured from.

Dataset	#Tracks	#Faces	#Subj
IMDB video faces	14,457	709,524	555
Men	7,104	341,721	208
Women	7,353	367,803	347
Accio [8]	38,464	N/A	121

Table 6.1: Summary statistics of the created database and the Accio database.

Chapter 7

Conclusion

We have proposed a method for automatic annotation of face sequences (face tracks) found in movie trailers. The method annotates face sequences with age, gender and identity. We exploited movie trailers and images of celebrities downloaded from Internet Movie Database (`www.imdb.com`).

We created a small-size manually annotated database that was used for tuning and evaluation of the proposed annotation algorithms. This database consists of face sequences tracked in 1,000 trailers. The manual annotation was assigned to the tracks in a web-based application we developed for this purpose.

The proposed method, tuned and evaluated on the manually annotated database, was used to create a large database of face sequences annotated by age, gender and identity. We named this database IMDB video faces. The created database consists of more then 14,000 tracks. The number of celebrities in the database is 555. The average number of tracks per celebrity is 26. It contains 709,524 facial images in total. The annotated age ranges from 15 to 79. Thanks to the evaluation on the manually annotated database, we estimate that 99.5% of tracks have correct annotation. To our knowledge, the created IMDB video faces is currently the largest database for age/gender prediction in terms of the number of celebrities contained. Moreover, the proposed algorithm can be readily used for processing more data which will be a subject of the future work.

Bibliography

1. ANTIPOV, G.; BACCOUCHE, M.; BERRANI, S.A.; DUGLAY, J.L. Apparent Age Estimation from Face Images Combining General and Children-specialized Deep Learning Models. In: *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2016.
2. ANTIPOV, G.; S.A.BERRANI; J.L.DUGELAY. Minimalistic CNN-based ensemble model for gender prediction from face images. *Pattern Recognition Letters*. 2016, vol. 70, pp. 59–65.
3. NIU, Z.; ZHOU, M.; WANG, L.; GAO, X.; HUA, G. Ordinal regression with multiple output CNN for age estimation. In: *In proc of CVPR*. 2016.
4. S.CHEN; C.ZHANG; DONG, M.; LE, J.; RAO, M. Using Ranking-CNN for Age Estimation. In: *In proc. of CVPR*. 2017.
5. PAN, Hongyu; HAN, Hu; SHAN, Shiguan; CHEN, Xilin. Mean-Variance Loss for Deep Age Estimation from a Face. In: *Proceedings of CVPR*. 2018.
6. ZHANG, Chao; LIU, Shuaicheng; XU, Xun; ZHU, Ce. C3AE: Exploring the Limits of Compact Model for Age Estimation. In: *CVPR*. 2019.
7. ROTHE, Rasmus; TIMOFTE, Radu; GOOL, Luc Van. DEX: Deep EXpectation of apparent age from a single image. In: *IEEE International Conference on Computer Vision Workshops (ICCVW)*. 2015.
8. GHALEB, Esam; TAPASWI, Makarand; AL-HALAH, Ziad. Accio: A Data Set for Face Track Retrieval in Movies Across Age. In: *Proc. of International Conference on Multimedia Retrieval*. 2015.
9. FRANC, Vojtech; CECH, Jan. Learning CNNs from Weakly Annotated Facial Images. *Image and Vision Computing*. 2018.
10. G.PANIS; A.LANITIS; N.TSAPATSOULIS; T.F.COOTES. Overview of research on facial ageing using the FG-NET ageing database. *IET Biometrics*. 2016, vol. 5, no. 2.
11. RICANEK, Karl; TESAFAYE, Tamirat. MORPH: A Longitudinal Image Database of Normal Adult Age-Progression. In: *IEEE 7th International Conference on Automatic Face and Gesture Recognition*. Southampton, UK, 2006, pp. 341–345.

12. GALLAGHER, A.; CHEN, T. Understanding Images of Groups of People. In: *Proc. CVPR*. 2009.

13. EIDINGER, Eran; ENBAR, Roee; HASSNER, Tal. Age and Gender Estimation of Unfiltered Faces. *Transactions on Information Forensics and Security (IEEE-TIFS), special issue on Facial Biometrics in the Wild*. 2014.

14. CHEN, Bor-Chun; CHEN, Chu-Song; HSU, Winston H. Cross-Age Reference Coding for Age-Invariant Face Recognition and Retrieval. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2014.

15. ESCALERA, S.; TORRES, M.; B.MARTINEZ; BAR, X.; ESCALANTE, H.J.; I.GUYON; M.OLIU; M.A.BAGHERI. Chalern looking at people and faces of the world: Face analysis workshop and challenge. In: *In IEEE CVPR Workshops*. 2016.

16. MOSCHOGLOU, S.; PAPAIOANNOU, A.; SAGONAS, C.; DENG, J.; KOTSIA, I.; ZAFEIRIOU, S. AgeDB: the first manually collected, in-the-wild age database. In: *Proceedings of IEEE Int Conf. on Computer Vision and Pattern Recognition (CVPR-W 2017)*. Honolulu, Hawaii, 2017.

17. AGUSTSSON, E.; TIMOFTE, R.; ESCALERA, S.; BARO, X.; GUYON, I.; ROTHE, R. Apparent and real age estimation in still images with deep residual regressors on APPA-REAL database. In: *12th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*. 2017.

18. ZHIFEI, Zhang; YANG, Song; HAIRONG, Qi. Age Progression/Regression by Conditional Adversarial Autoencoder. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

19. CENTENO, Iván de Paz. *Ipazc/Mtcnn* [online]. 2020 [visited on 2020-04-07]. Available from: `https://github.com/ipazc/mtcnn`.

20. ZHANG, Kaipeng; ZHANG, Zhanpeng; LI, Zhifeng; QIAO, Yu. Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks. *CoRR*. 2016, vol. abs/1604.02878. Available from arXiv: `1604.02878`.

21. VGG@OXFORD. *Ox-Vgg/Vgg_face2* [online]. 2020 [visited on 2020-05-13]. Available from: `https://github.com/ox-vgg/vgg_face2`.

22. *How Long Does the Average Hollywood Movie Take to Make?* [online]. 2018 [visited on 2020-05-12]. Available from: `https://stephenfollows.com/how-long-the-average-hollywood-movie-take-to-make/` Library Catalog: stephenfollows.com.

23. GUO, Yandong; ZHANG, Lei; HU, Yuxiao; HE, Xiaodong; GAO, Jianfeng. MS-Celeb-1M: A Dataset and Benchmark for Large-Scale Face Recognition. *CoRR*. 2016, vol. abs/1607.08221. Available from arXiv: `1607.08221`.

24. CAO, Q.; SHEN, L.; XIE, W.; PARKHI, O. M.; ZISSERMAN, A. VGGFace2: A dataset for recognising faces across pose and age. In: *International Conference on Automatic Face and Gesture Recognition*. 2018.

25. *Encode/H.264 – FFmpeg* [online] [visited on 2020-05-21]. Available from: `https://trac.ffmpeg.org/wiki/Encode/H.264`.

26. *Encode/H.265 – FFmpeg* [online] [visited on 2020-05-21]. Available from: `https://trac.ffmpeg.org/wiki/Encode/H.265`.