The Face Recognition Project

Report

Implementation and Analysis of Advanced Deep Learning Models for Face Recognition

Authors:

G. L. Huynh, C. Koschenz, L. Livdāne, and K. Tran

December 6, 2023

Contents

1 Context

This project aims to develop and analyze face recognition models for identifying individuals in images. It integrates advanced machine learning, in particular deep learning techniques, to create a face recognition system. The project makes use of publicly available datasets, including the "5 Celebrity Faces" dataset (Sec. 4.2) and the "PINS Face Recognition" dataset (Sec. 4.3), to provide a reliable foundation for training and testing the model.

1.1 Historical Perspective and Controversies

Research on face recognition started in 1960 with Woody Bledsoe, Helen Chan Wolf, and Charles Bisson teaching a computer to recognize human faces. Their approach involved building a database of pictures and the coordinates of facial features, such as the mouth and eyes, as well as distances like the width of these features. A computer would calculate a set of distances from submitted face pictures and return records of possible matches.

A more notable achievement was the automated face recognition system developed by the FERET program, led by the Defense Advanced Research Project Agency (DARPA) and the Army Research Laboratory (ARL) in 1993. This system was designed to support security intelligence and law enforcement personnel in field identification processes. Once the developed system performances were acceptable, three companies were founded to commercialize the automated face recognition technology. One of the first use cases was the identification of prisoners in Minnesota using their mugshots.

Until 1990, the facial recognition systems were based on photographic portraits of human faces. In the early 1990's, research on face detection systems raised interest. Researchers wanted to detect faces among other objects in photographs. The most well-known method was Principal Component Analysis (PCA), developed by Matthew Turk and Alex Pentland.

A more recent notable development in face recognition is Ukraine's use of Clearview AI technology to identify dead Russian soldiers. The families of the identified soldiers are then contacted to raise awareness of Russian activities in Ukraine.

Nowadays, face recognition systems are used worldwide by both governments and private companies for a wide range of applications, including smartphones, video surveillance, and robotics. Yet, this technology raises controversies, especially concerning data privacy. These systems can assist governments in tracking criminals as well as tracking ordinary citizens. The facial recognition system implemented in China is one of them and raises concerns not only about data privacy, but also about the practice of publicly shaming people for "uncivilized" behavior.

1.2 Technical Integration

The technical aspects of the project contain:

- **Data Utilization**: the project utilizes publicly available datasets, ensuring access to diverse facial images and identities for model training and testing.
- **Deep Learning**: advanced deep learning algorithms are employed to build and test sophisticated face recognition models capable of accurately identifying faces and measuring confidence levels accurately.
- **Model Development**: the project includes the development of a reliable face recognition model, which can be implemented in various applications, ranging from identity management to security.

1.3 Scientific Integration

The project contributes to the evolving field of facial recognition and deep learning, offering fundamental insights into state-of-the-art applications. It serves as a practical example of applying machine learning and deep learning techniques to real-world problems, emphasizing the importance of facial recognition in biometrics and its wide-ranging application. This work supports the need for further research and advancements in robust face recognition algorithms.[1–6] as well as addressing ethical considerations and privacy concerns.[7]

1.4 Economic Integration

Face recognition technology has a broad market range, with applications in multiple sectors, including:

- **Identity Management**: The technology can be applied in scenarios where secure identity verification is crucial, such as access control systems, immigration, and border security.
- **Security**: The project's outcomes can enhance the understanding of security measures in public places, airports, and sensitive facilities by accurately identifying individuals.
- **Business Operations**: Face recognition can be employed in various service applications, such as personalized customer experiences, targeted marketing, and access control for business operations.

1.5 Use Cases

1.5.1 Biometric Border Checks in Europe

Before 2022, border guards of the Schengen Zone did not have any central system to register the entries and exits of non-EU citizens. The Large IT Systems Agency (EU-LISA) has since developed an entry/exit system to record the biometric data of all non-EU nationals crossing the external borders of the EU. This system includes facial data. It helps guards to detect travelers attempting to use multiple identities.

1.5.2 Biometric Screening at Airports

In recent years, global airports have been increasingly adopting facial recognition technology to both enhance security measures and streamline the travel experience for passengers. A prominent example is London Heathrow Airport, which has embarked on a comprehensive implementation of biometric screening for travelers. Upon arrival, passengers have the option to register with the biometric system by having their facial features scanned and linked to their passport details. This data is stored with the deployment of encryption techniques. As these passengers traverse through security checks or board their flights, the facial recognition system matches their facial data live with the stored information, thus eliminating the necessity for manual identity verification. This has speed up boarding procedures and minimized waiting times, significantly elevating the overall travel experience.

2 Scope

1. Face recognition covers two types of recognition processes:
   a) **Face verification**: The process of determining whether a person is really who they claim to be. That is a 1:1 matching process where a live picture is compared with the person's pictures stored in a database.
   b) **Face identification**: The process of determining the identity of a person among many others. This is a 1:N matching process where a face is compared against a set of many pictures in a database.

   This project will focus on face identification.
2. **Model Development**: We develop a functional prototype of face recognition model using machine learning, in particular deep learning techniques capable of accurately identifying individuals in photos.
3. **Dataset**: Due to the scope of the project, face localization is not taken into consideration primarily. Our aim is to build a Deep Learning model that, given an input face image, e.g., a CV-photo, the model is able to correctly predict the identity. This assumes that the identity is represented in the training set labels used to train the model. As a result, we have chosen to proceed with the "5 Celebrity Faces Dataset"(Sec. 4.2) in a first step. In a second step we chose to use a subset of the dataset "PINS Face Recognition" (Sec. 4.3) which contains pictures from 105 celebrities, and the identity information. This subset is composed of pictures from 10 celebrities. In a final step, we chose to use the whole "PINS Face recognition" dataset.
4. **Testing and Validation**: The model's performance will be evaluated using a validation set of the respective dataset. The aim is to ensure that the model can both locate and accurately identify faces.
5. **Performance Measurement**: A measure of certainty (confidence) will be implemented for the identified identities to provide insights into the reliability of the model's predictions.
6. **Framework choice**: We are aware that the 2 libraries, TensorFlow and PyTorch, are popular in the deep learning community and are capable of delivering high quality models. However, we have chosen to use TensorFlow for the following reasons. First, TensorFlow is the most prominent among different deep learning libraries, and therefore, various documentations, introduction guidelines, and general support can be found conveniently. Moreover, TensorFlow provides a ready-to-deploy environment which is suitable for cloud integration, specifically with Docker and Kubernetes. Lastly, TensorBoard is an efficient visualization tool for model monitoring, which enables the

ease of debugging.

7. **Visualization**: Visualizations and demonstrations will be created to showcase the model's performance and its practical applications.
8. **Specific resources**: this project will utilize GPUs provided by Google Colab to facilitate computational requirements.

3 Methodology

For this project, the chosen methodology is progressive and consistent with the 3 milestones proposed by DataScientest for the modelling phase. Therefore, the chosen methodology consists of three steps illustrated in Fig. 3.1.

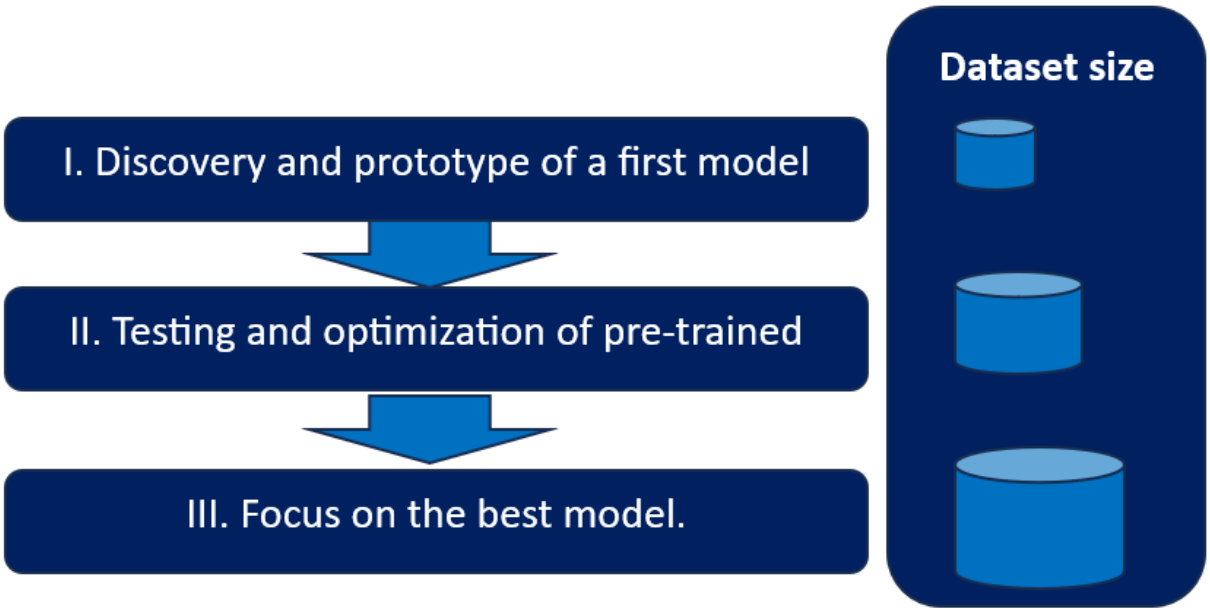


Figure 3.1: The project's methodology.

The first step is an opportunity to discover the complexity of deep learning, the frameworks TensorFlow and Keras, and to initiate modelling. The objective is to build a prototype that can download a data set of pictures, perform data augmentation, train the model, and perform inference, regardless of the accuracy (Chpt. 6). For this prototype we chose a small data set of pictures from the "5 Celebrities" Dataset (Sec. 4.2).

In a second step, the exploration of models is expanded by testing several models. The objective is to find the best model with the minimum loss and the best accuracy, knowing that over-fitting must be avoided. To fulfill this objective, the chosen models are trained with different sets of parameters (Sec. 4.3). Then the loss and the accuracy are plotted and compared (Chpt. 7). Each model is also tested in the inference mode to enable observations about its behaviour. The chosen data set would be adjusted considering initial observations from the previous step and expanded.

Finally, the best model will be explored more deeply with the full dataset based on that used for the second step to maybe open new perspectives (Chpt. 8).

4 Data Exploration and Visualization

4.1 The "Cats and Dogs" Dataset

The dataset "Cats and Dogs" can be found via the tutorial site of the Framework Keras [8]. It consists of images for the 2 classes, *cat* and *dog*. The details of this dataset are summarized in Tab. 4.1. The number of images for the both classes are uniformly distributed in the training as well as in the validation set.

Table 4.1: Details of the dataset "Cats and Dogs" for image classification used in [8]. Corrupted images where filtered out beforehand and the validation split ratio was chosen to 20%.

Feature	Description
Total Classes	2
Total Files	23410
Training Set	18728
▸ Training Cats	9420
▸ Training Dogs	9308
Validation Set	4682
▸ Validation Cats	2321
▸ Validation Dogs	2361
File Format	.jpg

4.2 The "5 Celebrity Faces" Dataset

The "5 Celebrity Faces" dataset is composed of 118 photos of five celebrities, having between 14 and 22 pictures each (Tab. 4.2). This dataset also includes a validation directory of 25 pictures equally divided between all the identities. The faces in the images come in various aspect ratios, positions, views (Fig. 4.3), and lighting conditions (Fig. 4.2). Some of the images contain two identities (Fig. 4.4), and some are in gray scale (Fig. 4.1). Therefore, although our dataset only contains a small number of pictures, it has a comparatively large number of diverse features.

This diversity of the dataset serves a dual purpose. On one hand, it helps in assuring the model's generalization capability. On the other, it presents a significant challenge during the validation phase if training examples are inadequate. This observation suggests a potential

Figure 4.1: Image types within the "5 Celebrity Faces" dataset, showing full-color (RGB) and gray-scale images.

Figure 4.2: Diversity of facial perspectives in the "5 Celebrity Faces" dataset.

Figure 4.3: Image framing variations, full-body versus. close-up shots, within the "5 Celebrity Faces" dataset.

Figure 4.4: Multi-face images from the "5 Celebrity Faces" dataset showing the complexity the model must handle in distinguishing multiple identities within a single frame.

Table 4.2: Summary of the "5 Celebrity Faces" dataset.[9]

Feature	Description
Total Files	118
▸ Training Set	93
▸ Validation Set	25
File Format	.jpg
Total Classes	5
▸ Ben Afflek	14 files
▸ Elton John	17 files
▸ Jerry Seinfeld	21 files
▸ Madonna	19 files
▸ Mindy Kaling	22 files

issue which we may encounter during the modeling phase. A possible solution would be to gather more 'extreme' training examples, or to augment the existing ones by rotating or even cropping them.

4.3 The "PINS Face Recognition" Dataset

Table 4.3: Summary of the "PINS Face Recognition" Dataset.[10]

Feature	Description
Total Files	17534
File Format	.jpg
Total Classes	105
▸ Files per Class	$95 < n < 225$ files

The "PINS Face Recognition" dataset [10] is composed of 17534 pictures from 105 celebrity identities. It is therefore significantly larger in comparison to the "5 Celebrity Faces" dataset of the previous section, but still manageable with respect to the resources available for this project.

The images are all face portraits, most of them are in RGB and only a small amount is in gray scale (Fig. 4.5). The position of the head can vary nevertheless. The subject does not necessarily face the camera all the time. Furthermore, the subject can sometimes close the eyes or wear glasses. There is good diversity in ethnicity and skin color. Moreover there is a good distribution in terms of gender. The only issue is that aged people are under-represented. We can only find Bill Gates and Morgan Freeman in that category.

To iteratively approach a higher number of identities recognized by our models we considered the complete dataset as well as a small subset of 10 selected identities (Tab. 4.4). This subset is considered for an intermediate explorative step before covering the complete dataset and will mainly be covered in Chpt. 7. This line of action allows us to select models with the

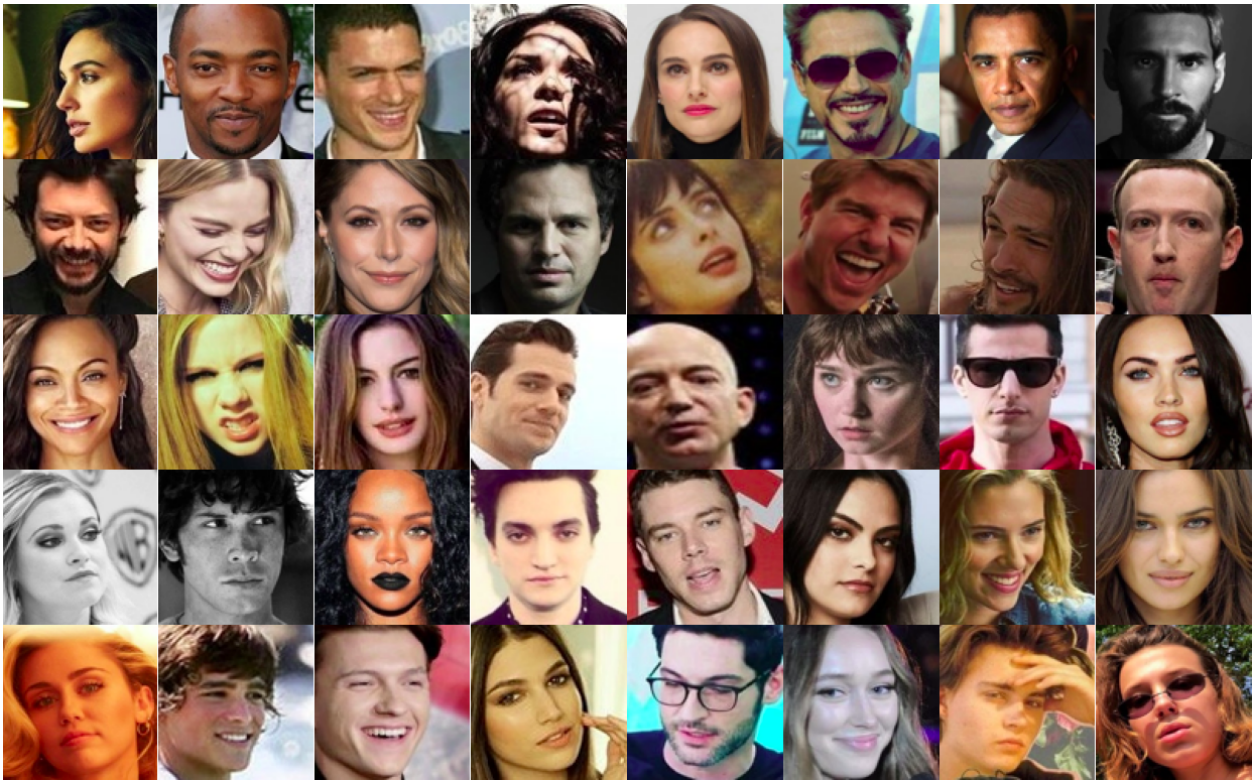


Figure 4.5: Samples of identities of the "PINS Face Recognition" dataset.

highest performance for final studies with lower usage of resources. For the selection of the subset we chose 2 main criteria:

- an equal number of identities per biological sex and
- for each sex, a broad diversity of the ethnicity and skin color.

Table 4.4: Summary of the dataset of the 10 selected identities of the "PINS Face Recognition" dataset.[10]

Feature	Descr.	Identities			
Total Files	1671	▸ Adriana Lima	213 files	▸ Chris Hemsworth	159 files
File Format	.jpg	▸ Rihanna	133 files	▸ Dwayne Johnson	141 files
Total Classes	10	▸ Scarlett Johanson	201 files	▸ Henry Cavill	195 files
		▸ Jason Momoa	184 files	▸ Shakira	154 files
		▸ Morgan Freeman	105 files	▸ Zoe Saldana	186 files

5 Data Preprocessing

Preprocessing of images is a critical step in preparing datasets for machine learning models. It has direct impact on performance and generalizability. This section covers the key aspects and techniques involved in image preprocessing used within this project, including reshaping and resizing images, enhancing their quality, performance optimization, and data augmentation strategies.

5.1 Image Reshaping, Resizing, and Enhancement

Reshaping and resizing images are pivotal steps in preprocessing, tailored to meet model requirements and optimize computational efficiency. This process involves altering the dimensions of images to fit the input size expected by specific neural network architectures. For instance, a convolutional neural network designed for a 256×256 pixel input will not effectively process a 1024×1024 pixel image without resizing. This resizing ensures uniformity in input data, which is crucial for consistent model training and evaluation. Techniques like bilinear or cubic interpolation are often used in this process, balancing the need for maintaining image integrity with computational efficiency.

Quality enhancement in image preprocessing focuses on improving the visual characteristics of images to improve model accuracy. This includes techniques like noise reduction, where algorithms filter out random variations in pixel intensity, and contrast adjustment, which alters the image to make key features more distinguishable. For instance, in facial recognition systems, enhancing contrast can make facial features more prominent, aiding in accurate identification.

5.2 Performance Optimization and Data Augmentation

Performance optimization in preprocessing involves scaling pixel values to a standard range, typically $[0, 1]$ or $[-1, 1]$, to aid faster and more stable training of neural networks. This normalization helps mitigate issues like the exploding gradient problem in deep networks, where large error gradients accumulate and disrupt the learning process. Additionally, preprocessing may include techniques like batch normalization, where input batches are standardized to reduce internal covariate shift, thus speeding up training and improving model performance.

Data augmentation addresses the issue of insufficient or inadequate data. Techniques include spatial transformation, like flipping, cropping, and rotation as well as color distortion and

information dropping methods such as “Cutout”, “Mixup”, “Cutmix”, and “Gridmask”. These augmentations enhance the dataset size and quality, aiding in model generalization and avoiding overfitting.

5.3 Implementation in Convolutional Neural Networks

The preprocessing steps are integrated into Convolutional Neural Networks (CNNs) when using TensorFlow. This includes automated image resizing using functions like `tf.keras.utils.image_dataset_from_directory` and image rescaling to address deep network training challenges. Additionally, image augmentation techniques are utilized during model training for regularization purposes, employing TensorFlow's layer types for flipping, rotation, and brightness/contrast variation. The data preprocessing steps mainly used within this projects are shown in Fig. 5.1.

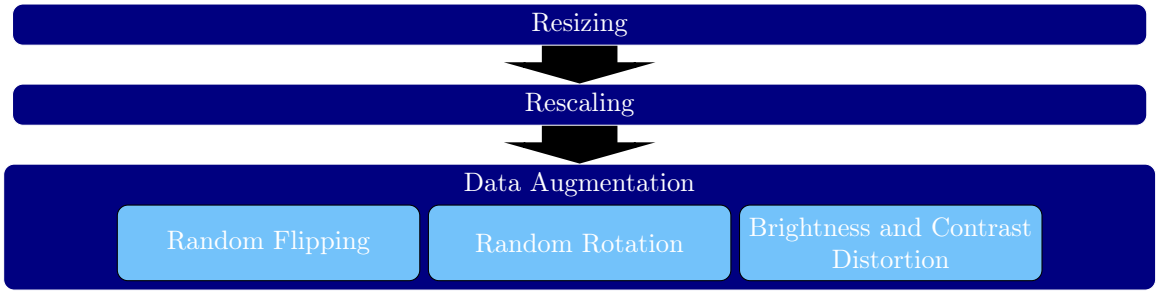


Figure 5.1: The data preprocessing process.

Image resizing and rescaling are integrated into the dataset import process and model input stages, respectively. Augmentation layers for flipping and rotation are activated only during training. Custom layers can be created for specific tasks like color distortion.

6 Initial Modelling

As a proof of concept and precursor for modelling and performance we decided to investigate the application of two models. At first, we followed the tutorial on the official Keras website introducing the classification of images from scratch using a small version of the Xception network and a dataset of pictures of "Cats and Dogs"(Sec. 4.1).[8]. After understanding the general approach and workflow, we proceeded with an implementation of the ResNet50 architecture with the "5 Celebrity Faces" dataset Sec. 4.2 to advance from a general classification problem to the specific case of face recognition.

6.1 Datasets

The "Cats and Dogs" dataset (Sec. 4.1) consists of two equally distributed classes, images of cats and dogs. It allows to study the binary classification problem for general object recognition.

For the "5 Celebrity Faces" dataset (Sec. 4.2) we expect the disadvantages to prevent the model from showing good accuracy or performance in general. The application is considered as a proof-of-concept and baseline for the following advanced modeling (Chpt. 7).

6.2 Models

We applied a small version of the Xception network (Sec. 7.2.4) [8, 11] to the binary classification problem of the "Cats and Dogs" dataset.

The ResNet50 model (Sec. 7.2.1) will be applied to the dataset "5 Celebrity Faces" in a following step. The architecture of ResNet and ResNetV2 is based on a residual learning framework which allows the training of substantially deeper networks. These residual networks are easier to optimize and can gain better accuracy from considerably increased depth.[12] Here we applied the unmodified implementation provided by Keras [13].

6.3 Results

The performance of the Xception model applied to the dataset "Cats and Dogs" becomes sufficient after 10 epochs (Fig. 6.1) and shows continuously increasing accuracy as well as decreasing loss with respect to the number epochs while the average gap between training

and validation data is decreasing for both measures. This trend indicates applicability of the model to this type of problem with low tendency to overfitting.

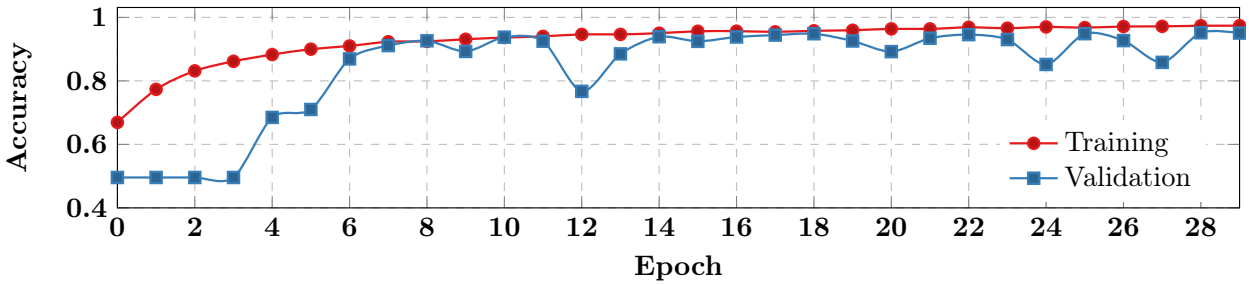
The application of the ResNet50 model to the dataset “5 Celebrity Faces” shows insufficient performance (Fig. 6.2), in particular in comparison with the solution for the general binary classification problem discussed previously (Fig. 6.1). While accuracy and loss for the training data converge against their upper respectively lower limit rapidly the corresponding measures for the validation data remain on a much lower plateau. This trend indicates a strong tendency of the model for overfitting the data and emphasizes the insufficient applicability.

6.4 Conclusions

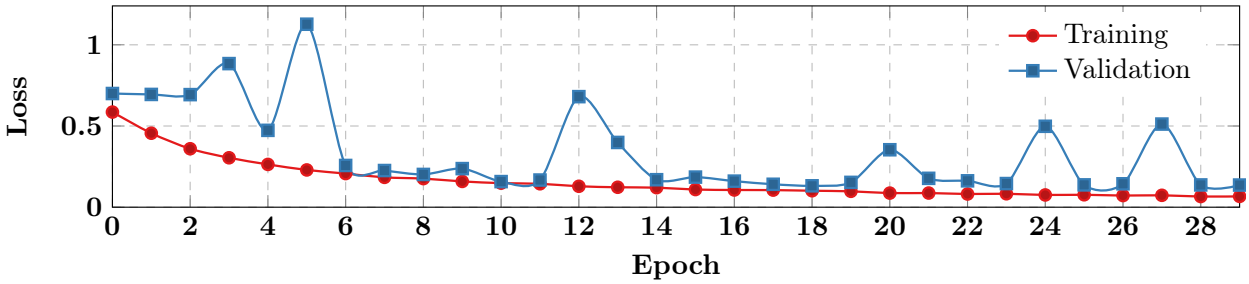
While the results of the binary classification problem for general objects shows a good performance with respect to accuracy and loss, the performance of the ResNet50 model applied to the dataset “5 Celebrity Faces” for face recognition shows insufficient performance and overfitting. The reasons for this lack of performance can be:

- small amount of images per identity,
- bias in significant features (see Sec. 4.2), and
- different types of images from full body to portrait images.

To circumvent these disadvantages, we decided to proceed with the larger dataset “PINS Face Recognition”(Sec. 4.3) of cropped faces in the following chapter.

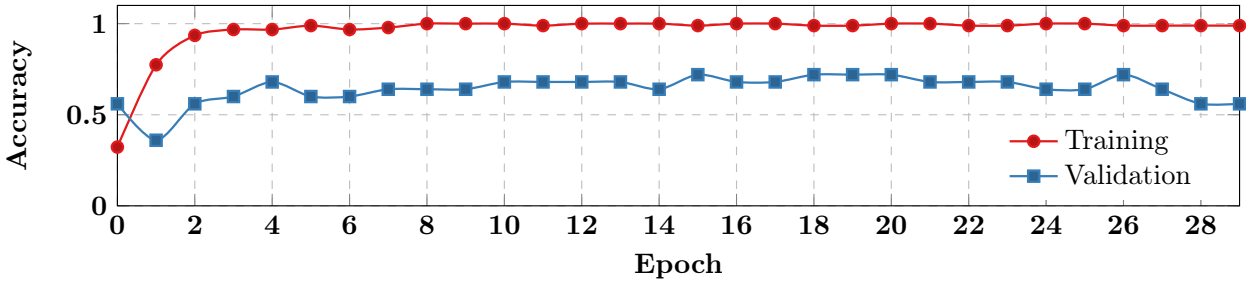


(a) Accuracy.

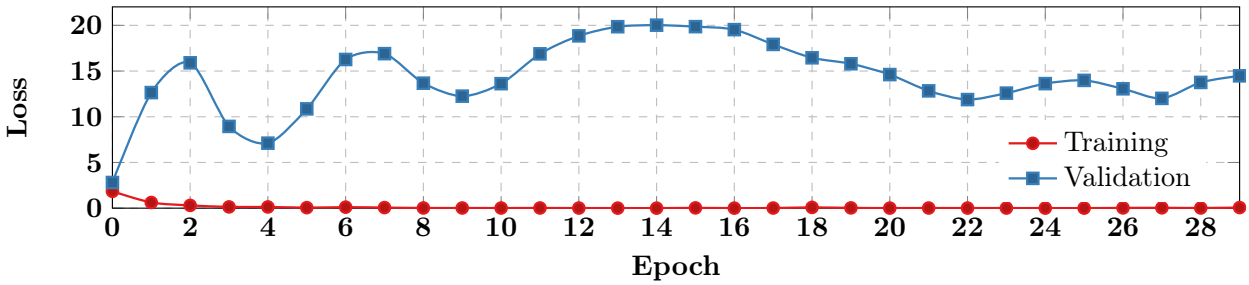


(b) Loss.

Figure 6.1: Overview of **(a)** accuracy and **(b)** loss of the Xception model applied to the dataset "Cats and Dogs" over 30 epochs following the tutorial in [8].



(a) Accuracy.



(b) Loss.

Figure 6.2: Overview of **(a)** accuracy and **(b)** loss of the ResNet50 model[12, 13] applied to the dataset "5 Celebrity Faces" over 30 epochs.

7 Advanced Modelling

In the following we distinguish between 2 main types of pretrained models, the models pretrained on the ImageNet dataset [14] or its subsets [15, 16] and the face-specifically trained models, like the FaceNet model [17] trained on the dataset *"Labeled faces in the Wild"* [18–20].

During the second phase of modeling, we tested several architectures with the following configurations:

- Training with new weights initialized with the He-initialization method [21].
- Fine-tuning with pretrained weights from ImageNet classification.
- Repeating the procedure with different parameters of choice.

7.1 The Dataset

We applied all models described in the following section to the dataset of 10 selected identities from the dataset "PINS Face Recognition" (Sec. 4.3). The focus on a constrained set of identities allows to obtain a first ranking of suitable models with lower ressource usage.

7.2 The Models

7.2.1 ResNet50

To leverage the power of deep residual learning for image recognition, we initiated our exploration with the ResNet50 model. The architecture of ResNet and ResNetV2 in general is based on a residual learning framework which allows the training of substantially deeper networks. These residual networks are easier to optimize and can gain better accuracy from considerably increased depth.[12] ResNet50 is known for its ability to train extremely deep neural networks using residual connections, mitigating the vanishing gradient problem. We applied this model optimizing it with a batch size of 64 and a target image shape of 160×160. Data augmentation techniques like color distortion, random flipping, and random rotation were employed.

7.2.2 Inception ResNet V2

Inception ResNet V2, which combines inception modules with residual connections, was chosen for its potential in complex image recognition. We evaluated this model adjusting batch sizes to 128, 64, and 32, with a consistent target shape of 160 × 160. Extensive data augmentation techniques, including color distortion, random flipping, and random rotation, were applied.

7.2.3 EfficientNet V2M

EfficientNet V2M[22], a model optimized for both accuracy and efficiency, scales network depth, width, and resolution in a balanced manner. We assessed its performance testing the batch sizes 64 and 32 and maintaining a target image shape of 160 × 160. Extensive data augmentation was applied as described in Chpt. 5.

7.2.4 Xception

The Xception model, with its depthwise separable convolutions, is designed for high-efficiency image classification tasks. Its architecture aims to entirely decouple the mapping of cross-channel correlations and spatial correlations in the feature maps of convolutional neural networks, which is a stronger version the hypothesis underling the Inception architecture.[11]

We evaluated it adjusting the batch sizes to 128, 64, and 32, with a consistent target shape of 160 × 160, and utilizing data augmentation of color distortion, random flipping, and random rotation.

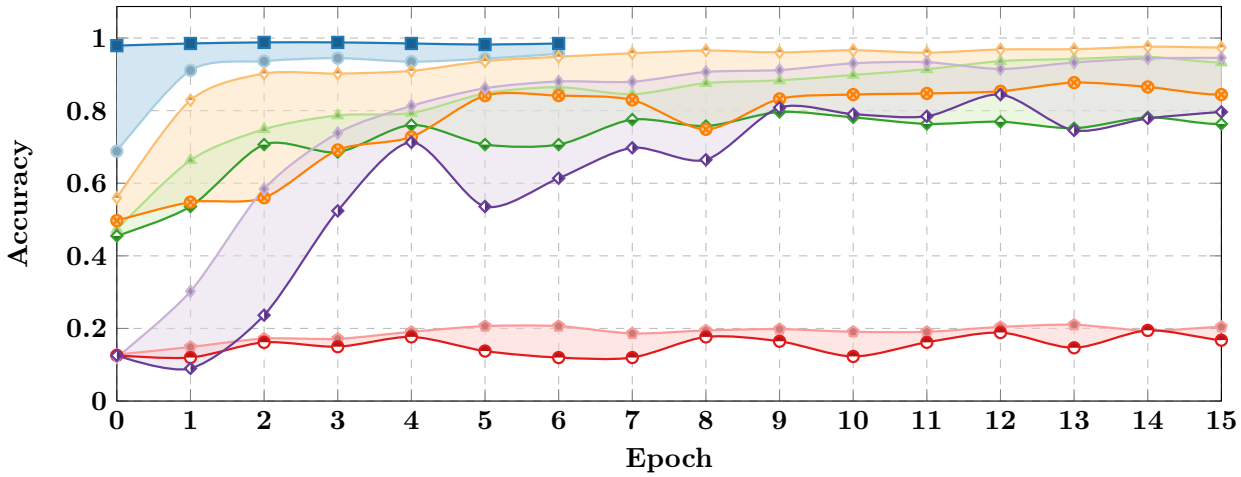
7.2.5 Facenet

In our exploration of advanced image recognition models, special attention was given to the Facenet model, with the latest implementation utilizing the `Inception_Resnet_V1` architecture. It should be mentioned that FaceNet is not actually the name of an architecture, but rather a group of pretrained CNN models that are used to tackle face recognition and/or verification tasks.

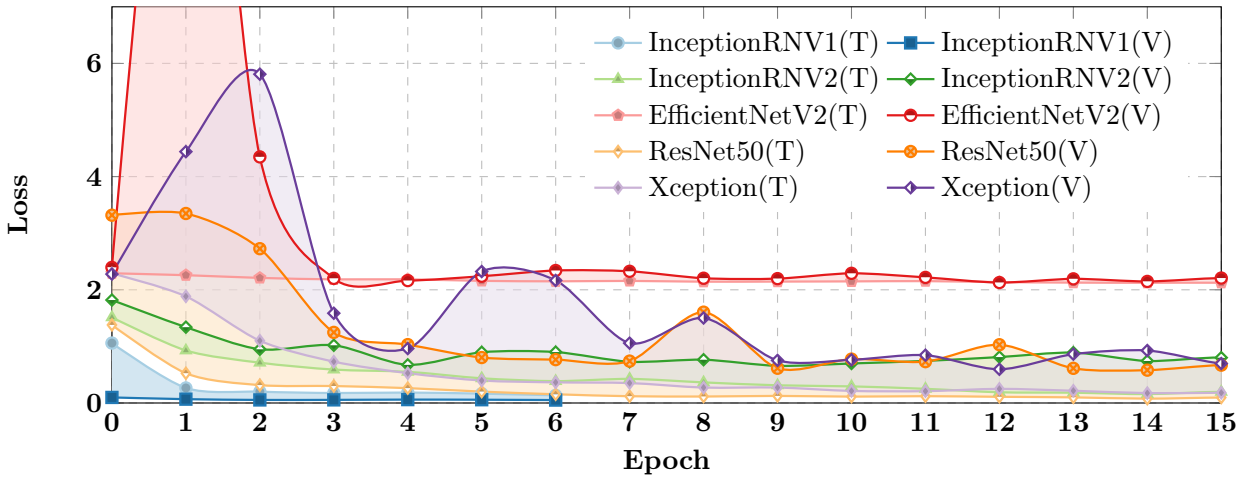
We standardized the input images with a target shape of 160 × 160 and applied data augmentation, using techniques like color distortion but excluding rotation. The batch size was set at 64. Fine-tuning was conducted on the `Block8_6_Conv2d_1x1` layer, which led to improvements in the model's ability to generalize.

7.3 Results

The application of the FaceNet model (Sec. 7.2.5) shows the best perfomance with respect to accuracy and loss as well as are very low tendency to overfitting (Fig. 7.1). This model also



(a) Accuracy.



(b) Loss.

Figure 7.1: Best performance with respect to **(a)** accuracy and **(b)** loss for the models described in Sec. 7.2. Lines and fillings are guides for the eyes with the filling indicating the gap between the models training and validation performance. The performance of the Inception ResNet V1 architecture ("FaceNet") is shown for 6 epochs to emphasize the rapid convergence and reaching sufficient perfomance limits (Chpt. 3).

shows the fastest convergence of accuracy and loss towards their upper respectively lower limit. The model achieved a training accuracy of 0.96 and exceeded this performance in validation accuracy, at 0.98. Both train and validation losses were low, at 0.13 and 0.15, respectively. The Facenet model outperformed other models and configurations tested, demonstrating robust learning without overfitting.

The remaining models, Inception ResNet V2, ResNet50, and Xception, show a much slower convergence of accuracy and loss as well as a finite tendency to overfitting (Fig. 7.1). The models pretrained with ImageNet obtained better results than those from models using random weights from He-initialization.

The ResNet50 model with fine-tuning on the `conv5_block1_out` layer achieved a training accuracy of 0.98 and a validation accuracy of 0.89. The training loss was 0.03, and the validation loss was 0.40, suggesting potential overfitting issues.

Using a batch size of 64, Inception ResNet V2 achieved a training accuracy of 0.97 and a validation accuracy of 0.81. The training loss was recorded at 0.09, while the validation loss was 0.77. These figures indicate a solid performance on the training data, yet the considerable difference between training and validation loss may point to potential overfitting issues.

The Xception model's best performance, with a batch size of 32 and fine-tuning applied to the `block13_sepconv1_act` layer, resulted in a training accuracy of 0.96 and a validation accuracy of 0.90, suggesting it can learn effectively from the data. However, the validation loss at this configuration was 0.50, indicating potential overfitting.

The model EfficientNetV2M shows insufficient perfomance over the whole range of 30 epochs. Accuracy as well as loss show a plateau behaviour, but most interesting also a low tendency to overfitting as can be concluded from the small gap between training and validation measures in comparison to the remaining models.

7.4 Conclusions

The results obtained from the ImageNet-models with fine-tuning are significantly better than those from models using He-initialization method, but still very mediocre regarding the original aim to deliver a robust and reliable face recognition model. The explanation is straightforward, given that ImageNet, which is a dataset for general-purpose object detection tasks[23], contains very few or even none of facial images.

The FaceNet family, on the other hand, is extensively trained on facial datasets e.g., VGG-Face2, CASIA-WebFace, MS-Celeb-1M, Youtube Faces DB, and LFW, and is therefore, a pivotal model for transfer learning and fine-tuning. Therefore we chose the Inception ResNet V1 architecture ("FaceNet") for extended training with the full dataset "PINS Face Recognition" in the next chapter.

8 FaceNet Model

The original FaceNet model was proposed by Google developers in 2015 to solve face verification problem on the Labeled Faces in the Wild (LFW) dataset and YouTube Faces DB [17]. The core principle behind FaceNet is the idea of projecting an image onto a hyper-dimensional vector space so that similar images, i.e., faces of the same identity, can be grouped together. This idea gives rise to the novel triplet loss function and a special sample selection strategy that focuses on generating "hard" triplets amid the training phase.

In Sec. 7.3, we show that this architecture yields by far the best performance with respect to accuracy and loss. Therefore we decided to test its performance on the full "PINS Face Recognition" dataset (Sec. 4.3). For our project, we use a recent version of FaceNet with Inception-ResNet-v1 architecture and pretrained weights from MS-Celeb-1M dataset[24].

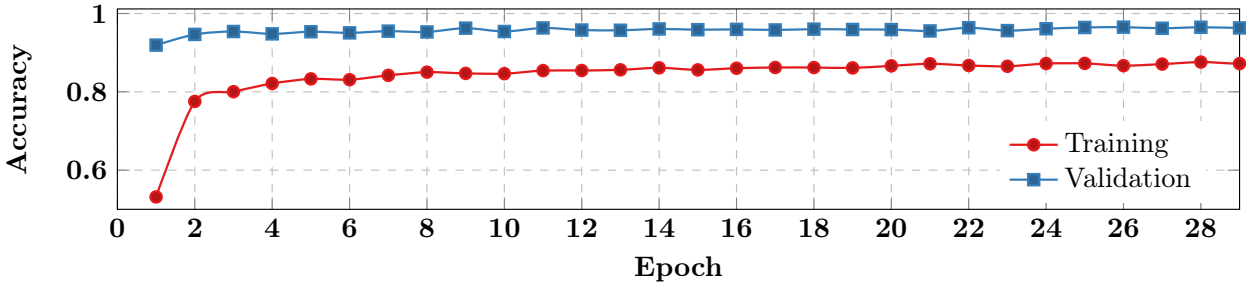
8.1 Inception-ResNet-v1

Choosing an appropriate architecture is crucial for a task that requires deep learning models. Unfortunately, this research field is yet to be mature, and a task-specific guideline is yet to be established. However, certain properties in recent deep learning models, e.g., Inception module and Residual block, allow for a better training phase as well as higher validation accuracy.

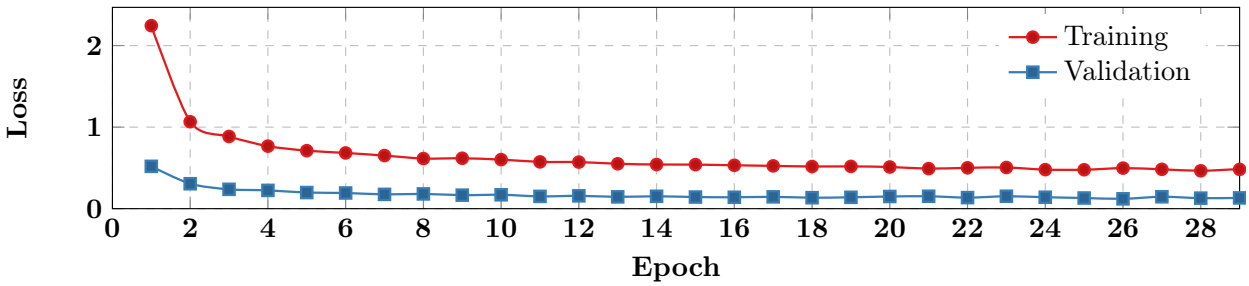
The theory of the residual block[25] states that the input of a convolutional block is concatenated with its output. In practice, this skip connection eases the learning of simple functions, e.g., the identity function, which is challenging for normal deep multi layer perceptrons. On the other hand, the Inception module employs 1×1 convolutions, resulting in a remarkable decrease in computational cost as well as an incorporation of different kernel sizes [26]. The fine-tuning of the architecture with respect to the dataset "PINS Face Recognition" was executed identically as in Sec. 7.2.5, in which we unfroze the network from the `Block8_6_Conv2d_1x1` layer onward.

8.2 Results

The fine-tuning result of FaceNet is illustrated in Figure 8.1, with an interesting observation. Both training and validation converge quickly after the first 5 epochs, yet there is a significant gap between them that even after a further 25 epochs, the loss in training seems unable to reach the desired value. This behavior indicates underfitting, and in fact, after eliminating

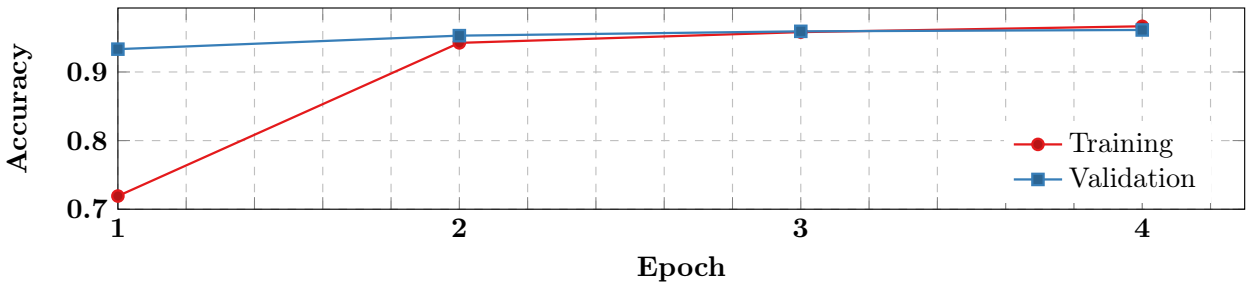


(a) Accuracy.

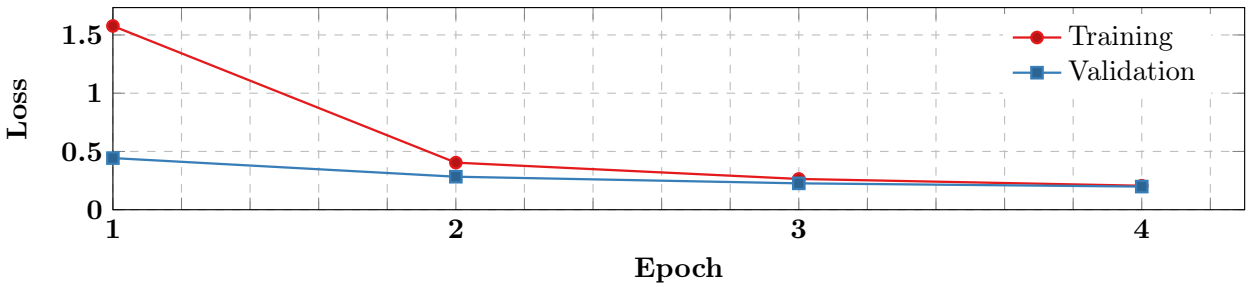


(b) Loss.

Figure 8.1: FaceNet fine-tuning results on dataset "PINS Face Recognition".



(a) Accuracy.



(b) Loss.

Figure 8.2: FaceNet fine-tuning results on dataset "PINS Face Recognition" without random rotation.

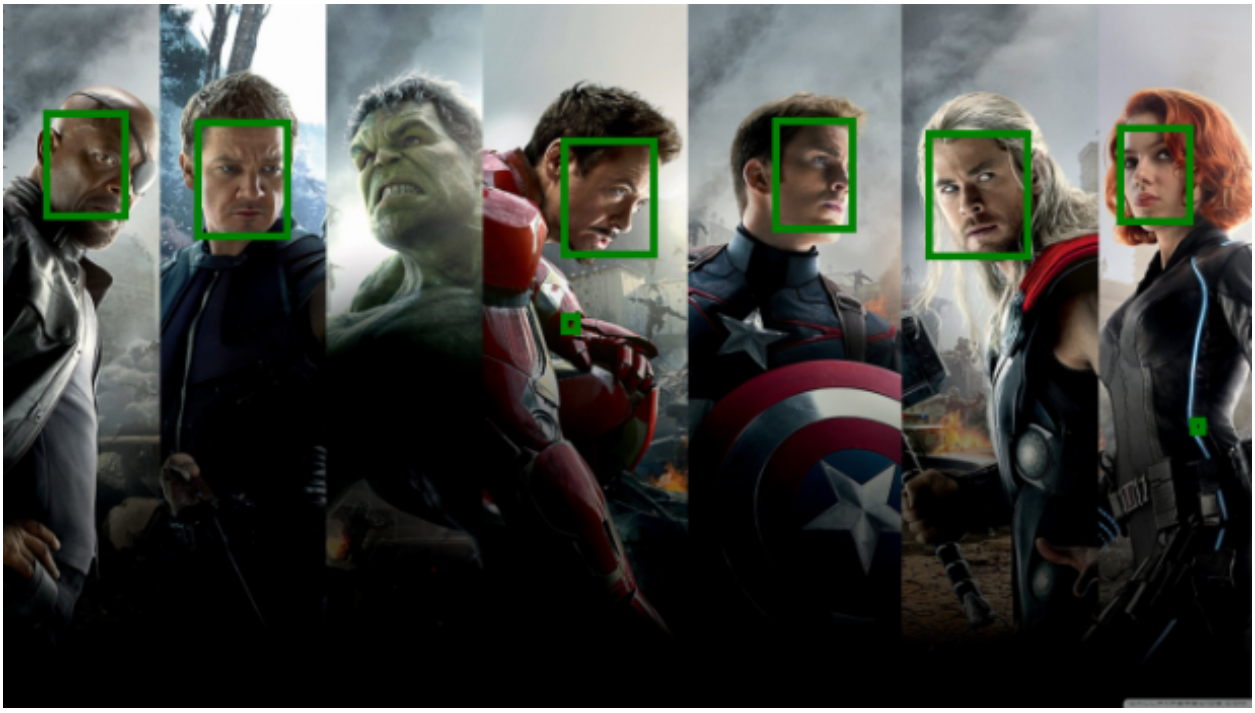


Figure 8.3: Face cropping using MTCNN.

random rotation from our image preprocessing pipeline, we are able to achieve excellent results in just 4 epochs (Fig. 8.2).

8.3 Perspectives

During the testing period with images that were not included in the dataset, we realized that the model performance is only reliable when predicting well-cropped facial images. This is no surprise, since FaceNet is trained particularly on facial dataset. Therefore, we decided to include a face-cropping module, MTCNN[27], into our inference pipeline. An example of face cropping using MTCNN is shown in Fig. 8.3. It should be noted that the module is not 100% reliable and can sometimes produce odd predictions. Furthermore, faces that are returned by MTCNN are too strictly cropped. As a result, we deliberately increase the sizes of the cropped faces by 10% of the returned width and height.

9 Model Interpretability and Further Considerations

(a) **(b)** **(c)**

Figure 9.1: Grad-CAM visualizations with the faces of **(a)** Ben Affleck and **(c)** Henry Cavill in **(b)** a scene taken from the movie Justice League in 2017. Both identities are correctly predicted by our model with over 95% confidence.

Since the introduction of advanced, powerful generation of computing resources, e.g., TPUs and GPUs, major breakthroughs in Deep Learning have also occurred in parallel. While the high-complexity characteristic of deep learning is required for its exceptional performance over traditional machine learning algorithms, the over-parameterized black-box nature prevents the comprehension of deep learning's predictive decisions. Indeed, the interpretability of deep learning itself is a branch of research.

In this section, we would like to mention a recent idea, Gradient-based Class Activation Map (Grad-CAM)[28], which serves as a special approach to highlight areas that a Deep Learning model deems important in a classification task. An implementation of Grad-CAM utilized in our project can be found under the repository in [29]. An example of Grad-CAM visualization on model predictions in conjunction with the face-cropping module MTCNN is given in Figure 9.1.

However, this should not be considered as a thorough conduct on the model interpretability and, therefore, is not a reliable justification of our model decisions. Further studies are required for stronger conclusions and gathering insights in the decision rules of our model for reliable interpretation. Approaches for this further interpretation would be the application of support vector machines, decision trees, and the study of the set of "eigenfaces" after training.

10 Results and Conclusions

This project aimed to develop and analyze face recognition models for identifying individuals in images. We started to build a prototype with Resnet50 one the models provided by Keras, and pre-trained with the ImageNet dataset. This model was trained on the dataset "5 Celebrity Faces" containing 118 images. The results were far from perfect but encouraging. That led us to explore several pre-trained models, with a slightly more extended dataset of 10 celebrities from the larger dataset "PINS Face Recognition". The results showed overfitting in all the cases and led us to test Facenet, a model based on `Inception_Resnet_V1`, and pre-trained with a large dataset of face images. This last model provides the best results with the least risk of overfitting. From that point, Facenet was selected and trained on the whole dataset "PINS Face Recognition" covering 105 identities of celebrities with between 86 and 226 images per identities. The model provided mixed results that could be significantly improved by deleting the random rotation from the pre-processing steps. Moreover, cropping the faces in the images with the MTCNN module helps to improve the performances significantly.

This technology constitutes a good basis for a product that will facilitate video identification for different industries like banking and public services. Nowadays, the eiDAS 2.0 framework is in development in EU. This framework, originally used to define the different types of electronic signatures and the digital identity standards in Europe in its first version, will be improved to facilitate the video identification procedure. Until now, this video identification requires that a human being performs the identification before issuing the identity. Soon, this process will be automated, assuming that the identity provider comply with the eiDAS 2.0 standards, that will be aligned with the Single Sovereign Identity standard that the EU would like to deploy.

From the business point of view, the final model based on Facenet can be used as a base for any use case of face recognition that requires a database of known faces, in access controls and identity management for instance. If we can feed the inference pipeline with live pictures taken by a camera, it will be able to identify known people on demand.

Bibliography

[1] I. Adjabi, A. Ouahabi, A. Benzaoui, and A. Taleb-Ahmed, Electronics **9**, 1188 (2020).

[2] P. Kaur, K. Krishan, S. K. Sharma, and T. Kanchan, Medicine, Science and the Law **60**, 131 (2020).

[3] L. Li, X. Mu, S. Li, and H. Peng, IEEE Access **8**, 139110 (2020).

[4] M. O. Oloyede, G. P. Hancke, and H. C. Myburgh, Multimedia Tools and Applications **79**, 27891 (2020).

[5] W. Ali, W. Tian, S. U. Din, D. Iradukunda, and A. A. Khan, Multimedia Tools and Applications **80**, 4825 (2020).

[6] V. Lakshmanan, *Practical machine learning for computer vision, End-to-end machine learning for images*, edited by M. Görner and R. Gillard, First edition, second release (O'Reilly, 2021), 463 pp.

[7] F. Boutros, V. Struc, J. Fierrez, and N. Damer, Image and Vision Computing **135**, 104688 (2023).

[8] F. Chollet, *Image classification from scratch*, (Nov. 2022) https://keras.io/examples/vision/image_classification_from_scratch/.

[9] D. Becker, *Dataset 5 Celebrity Faces*, (2017) https://www.kaggle.com/datasets/dansbecker/5-celebrity-faces-dataset.

[10] Kaggle, *Dataset Pins Face Recognition*, (2023) https://www.kaggle.com/datasets/hereisburak/pins-face-recognition.

[11] F. Chollet, 10.48550/arXiv.1610.02357 (2016).

[12] K. He, X. Zhang, S. Ren, and J. Sun, 10.48550/arXiv.1512.03385 (2015).

[13] Keras, *ResNet50 Documentation*, (Nov. 2023) https://keras.io/api/applications/resnet/#resnet50-function.

[14] ImageNet, *Imagenet full dataset source*, (2023) https://www.image-net.org.

[15] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, International Journal of Computer Vision (IJCV) **115**, 211 (2015).

[16] ImageNet, *Subset of imagenet dataset*, (2018) https://www.kaggle.com/c/imagenet-object-localization-challenge/overview/description.

[17] F. Schroff, D. Kalenichenko, and J. Philbin, in 2015 ieee conference on computer vision and pattern recognition (cvpr) (June 2015).

[18] G. B. H. E. Learned-Miller, *Labeled faces in the wild: updates and new reporting procedures*, tech. rep. UM-CS-2014-003 (University of Massachusetts, Amherst, May 2014).

[19] E. Learned-Miller, G. B. Huang, A. RoyChowdhury, H. Li, and G. Hua, "Labeled Faces in the Wild: A Survey", in *Advances in face detection and facial image analysis* (Springer International Publishing, 2016), pp. 189–248.

[20] Kaggle, *Labeled Faces in the Wild*, (2023) https://www.kaggle.com/datasets/jessicali9530/lfw-dataset.

[21] K. He, X. Zhang, S. Ren, and J. Sun, *Delving deep into rectifiers: surpassing human-level performance on imagenet classification*, 2015.

[22] Keras, *EfficientNetV2 Documentation*, (Nov. 2023) https://keras.io/api/applications/efficientnet_v2/.

[23] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, in 2009 ieee conference on computer vision and pattern recognition (2009), pp. 248–255.

[24] H. Taniai, *Keras-facenet*, (Nov. 2023) https://github.com/nyoki-mtl/keras-facenet.

[25] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, 2015.

[26] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, *Going deeper with convolutions*, 2014.

[27] I. de Paz Centeno, *Mtcnn*, (Nov. 2023) https://github.com/ipazc/mtcnn.

[28] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, International Journal of Computer Vision **128**, 336 (2019).

[29] S. Woof, *Grad-Cam++*, (Nov. 2023) https://github.com/samson6460/tf_keras_gradcamplusplus.