

ARTICLE



<https://doi.org/10.1057/s41599-022-01090-y>

OPEN

Facial recognition as a tool to identify Roman emperors: towards a new methodology

Darshan Srirangachar Ramesh¹, Sam Heijnen², Olivier Hekster², Luuk Spreeuwers¹ & Florens de Wit²

Portraits of Roman emperors are traditionally recognised by their unique coiffure patterns, a method that runs the risk of ignoring portraits that do not cohere to the standardised image of the emperor. This article investigates whether it is possible to recognise and distinguish emperors using the facial features of their portraits. By using a technique called transfer learning, it utilises existing deep-learning facial recognition models, augmented with images of Roman imperial portraits, to provide a new empirical foothold in the debate of Roman emperor recognition. The results of the experiments demonstrate that by only a limited amount of training, such a so-called “pre-trained” model (i.e., InceptionResnet-V1) is able to correctly classify most images in the dataset of Roman emperors. As such, this article has made a first step towards applying facial recognition models to the study of ancient imperial portraiture.

¹Data Management & Biometrics, Electrical Engineering, Mathematics and Computer Science, University of Twente, Enschede, The Netherlands. ²Radboud Institute for Culture and History, Radboud University, Nijmegen, The Netherlands. ✉email: d.srirangacharramesh@utwente.nl

Introduction

The Roman Empire was one of the largest empires the world has ever seen. For a long period of history, one man stood at the head of this enormous political entity: the Roman emperor. His presence was felt in almost every city in the ancient Roman world. In a letter to the Roman emperor Marcus Aurelius, the Roman rhetorician Marcus Cornelius Fronto writes that “you know that in all money-changers bureaus, booths, bookstalls, eaves, porches, windows, anywhere and everywhere there are likenesses of you exposed to view” (Fronto, *Ep.* 4.12.4. 207, transl. Haines, 1919). Indeed, portraits of the Roman emperors, carved from marble or cast in bronze, were omnipresent in the Roman Empire. Since the majority of the inhabitants of the Roman Empire would never get to see the emperor in person, statues and busts of the emperor played a crucial role as proxies in visualising imperial leadership (Ando, 2000; Stewart, 2003; Fejfer, 2008; Lahusen, 2010).

This article is concerned with the recognisability of Roman imperial portraits. It argues that the current method for identifying imperial portraits runs the risk of ignoring portraits that do not adhere to the standardised imperial image. In this article, we explore the potential of facial recognition software as an alternative tool to identify Roman imperial portraits. If facial recognition is able to match the effectiveness of the method currently used, it might provide researchers with a new empirical foothold. At the same time, applying a method that is designed for living human faces comes with its own set of methodological obstacles that must be solved before such a method can be used to recognise and identify Roman emperors.

Portraits of a Roman emperor were mainly spread via coinage and sculpture. The emperor’s face appeared *en profil* on the obverse side of coins together with his name(s) and titles. In sculpture, marble copies of the emperor’s portrait’s head were either placed on top of statues or presented in a shortened format: the portrait bust. In these different media, the emperor’s portrait generally followed a common set of characteristics. The Roman emperor Augustus, for example, is always presented with wavy locks of hair springing from his forehead that mostly follow the same pattern. This has everything to do with the production process of ancient sculpture; sculptors seem to have largely copied existing models (possibly in clay), which, in the case of imperial portraits, may have been distributed from the imperial centre (Fittschen, 1971; Pfanner, 1989). Art historians have long recognised this process and have developed a largely secure methodology, using coiffure patterns to identify Roman emperors among the bulk of ancient Roman portraits that have withstood the test of time (Fittschen, 2010).

Some scholars, however, have pointed out the pitfalls of the method described above, the biggest being that portraits that fail to demonstrate certain coiffure patterns are automatically excluded as imperial portraits (Riccardi, 2000). There is, then, a risk that portraits that do not adhere to the standardised image of an emperor are excluded from surviving corpora, leading us to recognise uniformity in the imperial image, where there may in fact have been much more variety in the emperor’s appearance. Indeed, recent years have also witnessed the discovery of so-called “unofficial” portraits of the emperor,¹ thereby forcing us to rethink the way in which we traditionally recognise and identify Roman emperors.

The debate has currently reached a standstill. What is needed is a new empirical foothold, which, on the one hand, builds upon accepted knowledge of the emperors’ appearances, but, on the other hand, refrains from using predetermined criteria such as coiffure patterns as prerequisites for successful identification. Here, we use data-augmented facial recognition software as an innovative tool to identify Roman imperial portraits. Recent decades have

witnessed the growing influence of modern technology in the study of ancient art and architecture (Schofield et al., 2012; Pollini, 2020), however, there have thus far been no attempts to utilise facial recognition software’s potential to the study of Roman imperial portraiture. Therefore, the main question this research article aims to answer is: can existing facial recognition methods based on Artificial Intelligence (AI) models be utilised to identify the portraits of Roman emperors?

Introduction to face recognition and deep-learning concepts and tools

Artificial Intelligence, Machine Learning and Deep Learning. Before discussing the details and results of our experiment on the faces of Roman emperors, it is worthwhile to briefly discuss the field of Artificial Intelligence (AI), particularly as applied to facial recognition. In the recent past, AI has made significant breakthroughs with regard to solving complex problems in Computer Vision (e.g., image classification) and Natural Language Processing (e.g., machine translation). In general terms, the field of AI contains Machine Learning (ML), which in turn envelops the field of Deep Learning (DL).

Machine learning (ML) is the technique to teach a machine to automatically learn concepts and knowledge from data, without being explicitly programmed to do so (Dargan et al., 2020). ML models, when trained with a sufficiently large dataset, can extract generalised features from the given dataset and provide reliable predictions on similar new data. Deep learning (DL) is a subset of machine learning. The key difference between ML and DL is that “classical” ML models tend to use existing and/or “engineered” features to perform a certain task (e.g., image classification), while DL models learn to automatically extract features from the input data *and* perform the task at the same time. DL models’ multiple layers of non-linear processing units are trained on a large amount of data to extract meaningful representations of the dataset (images, words etc.). The deeper into the layers one looks, the more abstract and dataset-specific the data-representation becomes. Such “learned” representations could be considered as a compact or compressed version of the originals, containing strong cues about the properties of the input data. Such cues are useful to perform some advanced tasks, e.g.:

1. Object detection (Zaidi et al., 2021)—Detection of objects of interest in images.
2. Image segmentation (Minaee et al., 2021)—Pixel level information about the object in images.
3. Image reconstruction (Liu et al., 2018)—Filling the corrupt parts of the image.
4. Image colorization (Zhang et al., 2016b)—Bringing old grayscale images to life by converting them to colour.

Convolution Neural Networks (CNNs). Convolutional neural networks (LeCun et al., 1998) are a popular technique in deep learning especially in the sub-field of Computer Vision. We follow Guo et al. (2016) in their explanation of how a CNN works. In Fig. 1a the general architecture of a CNN is displayed, showing the typical three layers i.e.: (1) convolutional layers, (2) pooling layers and (3) fully connected layers.

As can be seen in Fig. 1b, in convolutional layers the input (image or feature map) is “scanned” using filter kernels, which results in a single output per kernel per scanned position on the input. The filter kernels can be seen as representing particular patterns, so by applying them to the input image the resulting, so-called, feature maps represent the patterns that exist in the image. Multiple kernels give multiple, different, feature maps, that can be

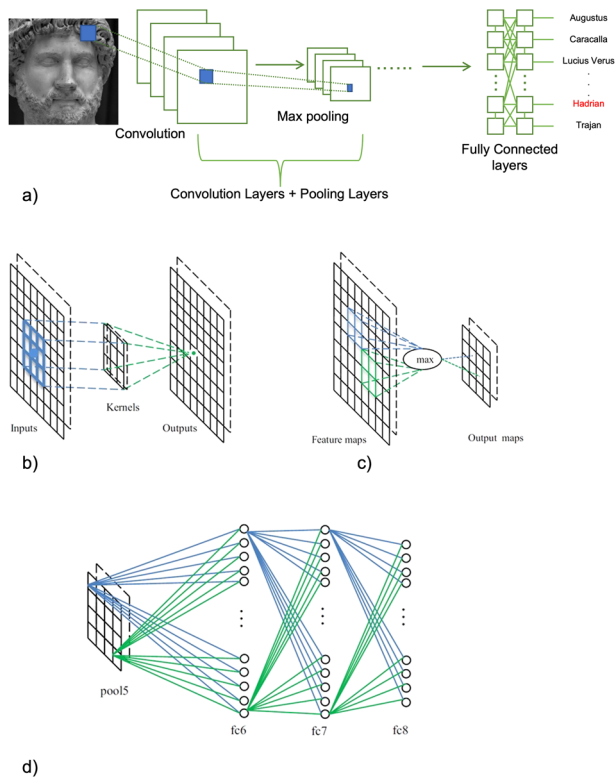


Fig. 1 CNN architecture and operation. **a** General CNN architecture. Image of Hadrian: © Centre for Art Historical Documentation, Radboud University. **b** Convolutional layer operation. **c** Max pooling layer operation. **d** Fully connected layer operation. **b**, **c** and **d** reprinted from Guo et al. (2016) with permission from Elsevier.

up to the size of the original input if the kernels are applied to every input position. By applying kernels to the two-dimensional image, the network has not just information on what patterns are present, but also on their special relationships, and it can learn a representation of the structure of the image.

A *pooling layer* (see Fig. 1c) is used to reduce the dimensions of the feature maps (and consequently the number of parameters necessary) by subsampling them. In the figure a 2 × 2 set of input values is reduced to a single value by taking the maximum. This is what is called max pooling, and is one of the commonly used pooling strategies next to average pooling.

The *fully connected layers* are what links a CNN with a “normal” artificial neural network. In a CNN these layers usually are situated at the end of the network and perform the classification task. They serve as the conversion of the two-dimensional feature maps into a one-dimensional output vector giving the classification. In Fig. 1d, one can see that in these layers all units are connected to all the previous layer’s units; hence “fully connected”.

The training of CNNs occurs in two stages: (1) forward pass and (2) backward pass. During the forward pass, the layers extract features from the given image in such a way that the earlier layers learn the low-level generalised features (e.g., edges) and the later layers must learn more complex dataset-specific semantic features (e.g., faces) (Yosinski et al., 2014; Du et al., 2022). The output predictions are then compared with the ground truth label to compute the loss. A good loss function (for classification) will cause the network to group visually similar objects together, while it separates dissimilar objects by a wide margin. The calculated loss values are then used in the backward pass, which tries to reduce the loss value (in the next forward pass) by adjusting the network

weights and biases based on how the loss function depends on them, i.e., the gradient of the loss function. As the adjustments are calculated by moving “backwards”, this is called “Backpropagation”; since the loss value is pushed down through the gradient this technique is called “gradient descent”. After a backward pass the network parameters are used for the next forward pass. The network learning process is completed after a series of sufficient iterations of forward and backward passes. Once the network is trained, it is able to summarise the important features within the images, regardless of their location (Wang et al., 2020).

Facial recognition. Facial recognition (FR) is used as a key tool to perform various biometric tasks on human faces, such as verification (are these images the same person?) and identification (who is this person?). In general, a FR system consists of three steps:

1. Face detection and alignment;
2. Feature extraction;
3. Face recognition.

The current face recognition algorithms achieve accuracies at the human-level (e.g., Phillips et al., 2018), which may be due to the application of the aforementioned CNNs to perform these three steps, either using multiple separate dedicated networks or in an “end-to-end” fashion, in a single network.

Challenges. Before applying facial recognition to Roman imperial portrait recognition, we have to address several issues. First, when compared to real human faces, the portraits of Roman emperors lack important features such as texture and colour. These features are very likely used by existing face recognition systems to distinguish between real human faces. Ideally, we would like to train an existing face recognition algorithm to extract specific features of imperial portraits, i.e., ignore textures, colour and any other feature that doesn’t appear in imperial portraits. However, as said, most Facial Recognition (FR) systems available for research use Deep Convolution Neural Network (DCNN) models. Since such models require thousands of images to learn meaningful representations, training a face recognition DCNN model from scratch specifically for recognising imperial portraits is impossible because there is insufficient data available: only ca. 2100 imperial portraits have survived. As a solution, we can base a specific model on an existing facial recognition model (i.e., already trained on the images of human faces), which is then “modified” by partially training the network using the images of imperial portraits so it learns to recognise the “faces” of emperors as they are represented in sculpture; this technique is called “transfer learning” (Zhuang et al., 2020) because it transfers the model’s skill on a more general task—in this case “real” face recognition—to a specific related problem—i.e., Roman imperial portrait recognition.

Second, though some scholars have tried to set specific criteria or guidelines for photographing imperial portraits (e.g., Fittschen, 2010) there is not only a large variation in physical conditions of the surviving material, but also in their photographic documentation, which poses an additional set of challenges for face recognition models trained on real human faces, such as:

- Pose and lighting of the subject can influence the performance of face recognition even for real human faces. In Fig. 2a/b it is clear there is significant variation in both aspects in the dataset of imperial portrait images (even just for emperor Hadrian).
- Many portraits have been damaged (see Fig. 2c), and, as a consequence, also may contain modern restorations.

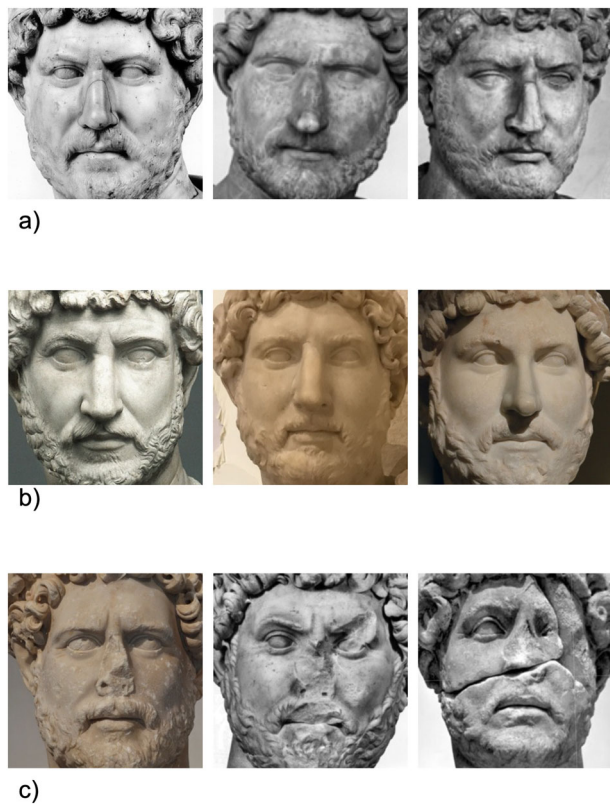


Fig. 2 Images of Emperor Hadrian in the dataset. **a** Varying poses. **b** Varying lighting conditions. **c** Damages in the face areas. Images A1-A3, B1 and C2-C3 by the Arachne object database (licensed under CC license BY-NC-ND 3.0), B2 and C1 by Carole Raddato (licensed under CC BY-NC-SA 2.0), and B3 authors' own.

This type and level of distortion and/or modification, makes these portrait images much more dissimilar to other images of imperial portraits, indifferent whether of the same emperor or not. In this the portraits are quite different from the real human faces used for training the face recognition model; though cosmetic surgery might cause a similar effect as restorations, a severe face injury similar to the damage is rare and will not be taken into account for automated face recognition.

- In contrast to real human faces, Roman imperial portraits are more or less idealised renderings of the emperor's face based on available prototypes that display the "standardised" or "accepted" view—at different times and/or in different settings—rather than a realistic depiction of what the emperor looked like. On the other hand, sculptors would vary in their skill of reproducing the portrait from the prototype. Portraits of the same emperor can thus vary greatly depending on the availability of prototypes and the skills of the sculptor.

The proposed method

We propose to use existing "pre-trained" DCNNs and apply transfer learning using photographs of imperial portraits to make them fit-for-use on imperial portrait recognition. A visual outline of the proposed method is shown in Fig. 3.

First, we use a pre-trained CNN-based model (Multi Task CNN) from Zhang et al. (2016a). This method is based on a cascade of three networks; The Proposal network (P-net), which generates candidate windows, the Refinement network (R-net), which refines those proposals, and the Output network (O-net),

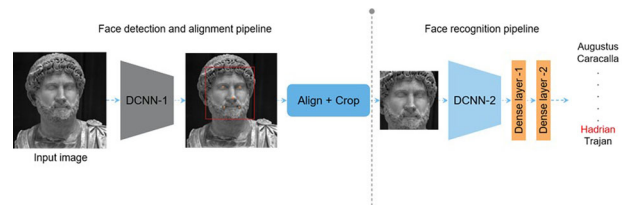


Fig. 3 Proposed imperial portrait recognition pipeline. Image of Hadrian: © Centre for Art Historical Documentation, Radboud University.

which produces the final bounding box and facial landmark position. For further details on the network architecture and training we refer to Zhang et al. (2016a).

We use this method to detect faces, find a bounding box and facial landmarks and use that data to crop, align and resize the faces to a standard size of 256×256 pixels, whilst maintaining the original aspect ratio. Images are then saved in a separate database.

After this pre-processing step we will apply transfer learning to fit the Inception-ResNet-v1 model to our data. The Inception-ResNet network architecture combines the so-called "inception" units with residual connections, which speeds up training and improves performance with respect to similar Inception networks (Szegedy et al., 2017).

Dataset. For the purposes of this study, we have set up a dataset of 673 images, consisting of 645 images of nine emperor classes (Augustus, Trajan, Hadrian, Antoninus Pius, Marcus Aurelius, Lucius Verus, Commodus, Septimius Severus, and Caracalla), and 28 random images of non-emperors that serve as a distractor class to the model. The images were collected by using existing online datasets, catalogues, and photographs we made ourselves. The dataset includes images with varying poses, illumination, and damages on different areas of the faces. In case so-called childhood portraits of the emperors mentioned above have survived, these are included in the dataset as well.

These 673 images were subsequently split into a training set (of 351 images) and a validation set (of 174 images) as seen in Fig. 4a. The validation set is used during training to assess whether the model is actually learning general features of the images or is just memorising the data; the latter is called overfitting.

The number of images per emperor is not the same (i.e., a class imbalance). To avoid that the model is biased towards the class with the most images it is important to preserve the relative size of the classes between training and validation set. This is why we used stratified splitting instead, where a fixed fraction is randomly sampled from each class rather than from the entire dataset. This results in same subdivision between classes, within training and validation sets, as visible in Fig. 4a.

We have also set up an additional test set of 148 randomly selected imperial portrait images (Fig. 4b), to examine if the model has actually gained the ability to infer the class of (relatively) unseen data after training is done. The test set is entirely independent of the aforementioned training and validation data; there is no data leakage from either the training or validation set into the test set. Note that Fig. 4b shows this dataset has a different class distribution as the training and validation sets for few classes due to the unstratified random selection. Because of the limited number of non-emperor portraits we have decided to not include any in the test set. The aim of this article is to establish if we can identify which Roman emperor we are dealing with through face recognition software, rather than signalling whether this is an emperor or some other citizen or administrator.

As the given number of imperial portrait images available is rather small—considering the network was trained on millions of

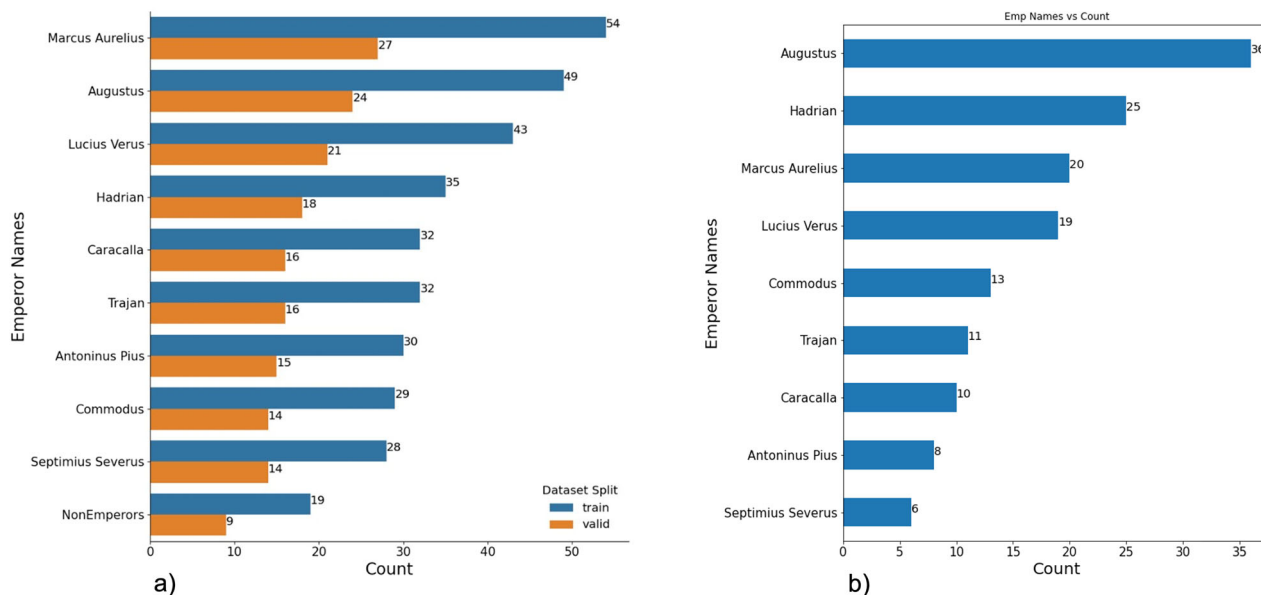


Fig. 4 Dataset information. **a** Training, validation and **b** testing sets. As our dataset is imbalanced, the dataset is split in a way that the training and validation datasets have similar subdivision between emperors and the non-emperors class. Testing set is chosen randomly and is not split in a stratified manner for some classes.

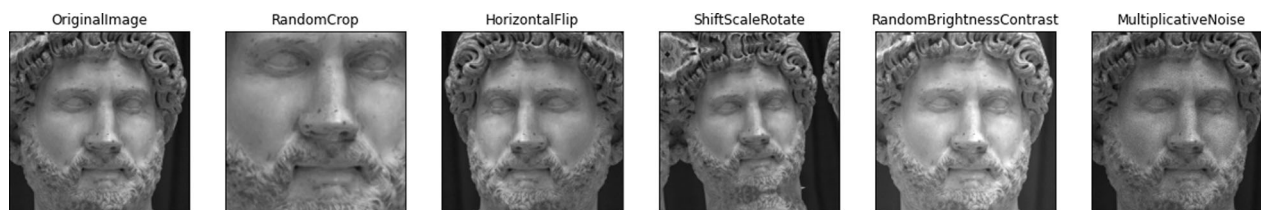


Fig. 5 Augmentations applied to images (Buslaev et al. (2020)). Image of Hadrian: © Centre for Art Historical Documentation, Radboud University.

Table 1 Data augmentation parameters used in our experiments.

Augmentations	Parameters	Probability
Horizontal flip	-	0.5
Shift scale rotate	Shift limit = 0.02 Scale limit = 0.2 Rotate limit = (-10,10) deg	0.8
Random crop	Size = (224,224)	0.8
Random brightness contrast	Brightness limit = 0.2 Contrast limit = 0.2	0.8
Multiplicative noise	Multiplier = (0.8, 1) Per channel = True	0.8

images—this limits our ability to adjust the model to work effectively with those portraits. A way around this, commonly used in deep-learning models for computer vision, is the use of so-called augmentations; i.e., applying modifications to the existing images—e.g., horizontal flipping, adjusting brightness and contrast and shift scale rotating (see Fig. 5/Table 1)—to increase the number of available images for training. This strategy effectively multiplies our training dataset size by six, with the added value that the model can deal with a variety of challenging conditions in poses and lighting.

Experiments: re-training of the classifier network

We have performed two experiments using the above mentioned InceptionResnet V1 CNN model trained on the VGGface2

dataset (Timsler, 2019). In Experiment 1, we have replaced the last two, densely connected, layers and re-trained them by using our data. This was necessary because (1) the original network classifies into 1000 classes while our dataset only has 10, and (2) we want the network to select only relevant features from the collection it has learned to extract from real facial images (e.g., ignore features derived from skin colour). In Experiment 2, we used the model resulting from Experiment 1, but this time trained the specifically the last convolution layer of the feature extractor. Usually, the higher layers of the network have learnt the class-specific features—e.g., entire face, their poses etc.—from the dataset (Zeiler/Fergus, 2014). So we re-train the last convolutional layer to extract the most dataset-specific information related to the imperial portrait classes rather than to face images in general. Figure 6 shows the different steps of the two experiments, using an example of a portrait of Hadrian (Sevilla, Museo Arqueológico) in our validation set.

Training and hyperparameters details. During training, all the images of size 256 × 256 are resized and they are randomly cropped to get a 224 × 224 image size 80% of the time, which are further resized to 160 × 160. Image normalisation is done such that an image’s value is between -1 and 1. The image size of 160 × 160 is required because the model used was trained on this image size. Details on training and hyperparameters are given in Table 2. We have used PyTorch (Paszke et al., 2019) as our deep-learning framework.

We use the F1-score and the confusion matrix to measure how well the models does on the validation and test set. The confusion

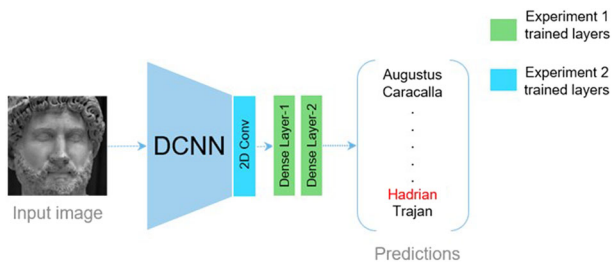


Fig. 6 Schematic overview of experiments. Experiment 1 trains densely connected layers of Inception-Resnet V1 model. Experiment 2 trains last 2D convolutional layer from experiment 1 model. Image of Hadrian: © Centre for Art Historical Documentation, Radboud University.

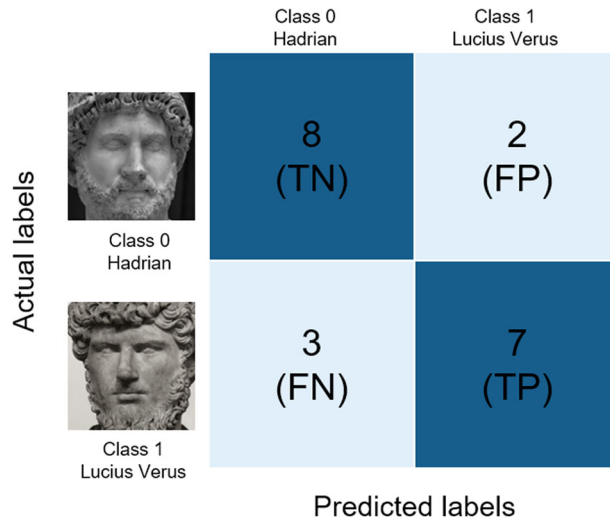


Fig. 7 Confusion matrix for a simple binary classifier. As an example a simple binary classifier’s performance is measured here where the model correctly classifies eight out of ten images of Hadrian but also identifies two images as Lucius Verus. Same analysis applies to the class Lucius Verus. TP = True Positive. FP = False Positive. TN = True Negative. FN = False Negative. Images of Hadrian and Lucius Verus: © Centre for Art Historical Documentation, Radboud University.

Table 2 Details on training and hyperparameters.

Parameter	value	Additional details
Dropout rate	0.4	Before dense layer 1 (see Fig. 6)
No. of epochs	40	
Batch size	Train: 16	Softmax activated
	Validation: 8	
Loss	Cross-entropy	Softmax activated
Learning rate	Type: multi step	
	Initial: 0.005	
	Step: 10%	
Optimiser	At: epoch 5 and 35	Adam

matrix for a simple binary classifier model is shown in the Fig. 7. In a confusion matrix the numbers on the diagonal line represent the correct predictions—i.e., True Positives (TP) and True Negatives (TN)—while the non-diagonal elements represent false positives (FP) and false negatives (FN).

Using these numbers, the F1-score can be defined as:

$$F1\ score = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

The F1-score is 1 when the prediction is perfect (i.e., no false positives or negatives) and 0 when no positive prediction is correct.

Results

Experiment 1 is the traditional way of transfer learning using a DCNN model. With this experiment we test if face recognition can indeed be applied to our set of Roman emperor faces, despite of the challenges discussed above. As such, the results of this experiment provide a baseline model. Figure 8a/c show the confusion matrices on the validation and test set of this model respectively and Table 3 gives the F1-scores per class on both sets. Apparently the model is able to still correctly classify most images despite the lack of texture and colour in the “faces” and the presence of (restored) damage, but there is room for improvement.

Experiment 2 refines the baseline model from experiment 1 by training the most specific part of the feature extractor. This is therefore a fine-tuned model for this particular dataset and problem. Figure 8b/d and Table 3 give the results of this model on all classes separately. To get an overall estimate of model performance, we have taken a weighted average of the F1-score per class with the number of images per class as weights; note this is a different averaging method as macro or micro averaging commonly used with F1-scores.

The weighted average values of the F1-score in Table 3 show that the fine-tuned model indeed improves on the baseline on both validation and test set, outperforming it by 0.03 on validation (F1: 0.95) and 0.09 on the test set (F1: 0.90).

Another way to visualise the results of Experiment 2 is by projecting the model’s embeddings in a two-dimensional space, using a dimensionality reduction technique called UMAP (McInnes/Healy/Melville, 2018). From the figure below (Fig. 9), it becomes clear that the images from the ten different classes in our training set (Fig. 9a) are nicely grouped together, which means that the model is able to distinguish between images of different classes (small intra-class and larger inter-class distance). The integrity of these clusters is largely maintained when projecting the data from our validation set (Fig. 9b) as well but with a few noteworthy exceptions. Specifically when it comes to our distractor class of non-emperor images, a wider dispersion, sometimes overlapping with the different emperor classes, can be observed. This is not that surprising considering that this class is in itself not a coherent set of images of the same person.

Discussion

Though the performance of our method is as to be reasonably expected, it is not up to the level of state-of-the-art face recognition systems; our Experiment 1 system has achieved an accuracy of 81.1% and our Experiment 2 an accuracy of 89.2%, which is still significantly lower than the low 99% accuracies most current DCNN face recognition networks achieve. It is, however, still quite remarkable that the quite low number of images that are available still allows for a “custom” system for imperial portrait recognition.

With regard to the misclassified images in the validation and test sets, it is important to emphasise that some misclassifications may be the consequence of a phenomenon inherent to the medium of imperial portraiture, namely that emperors with some regularity assimilated their public image so close to each other that it is near-impossible to tell who is who. This phenomenon (*Bildnisangleichung*) is well attested for emperors like Caracalla

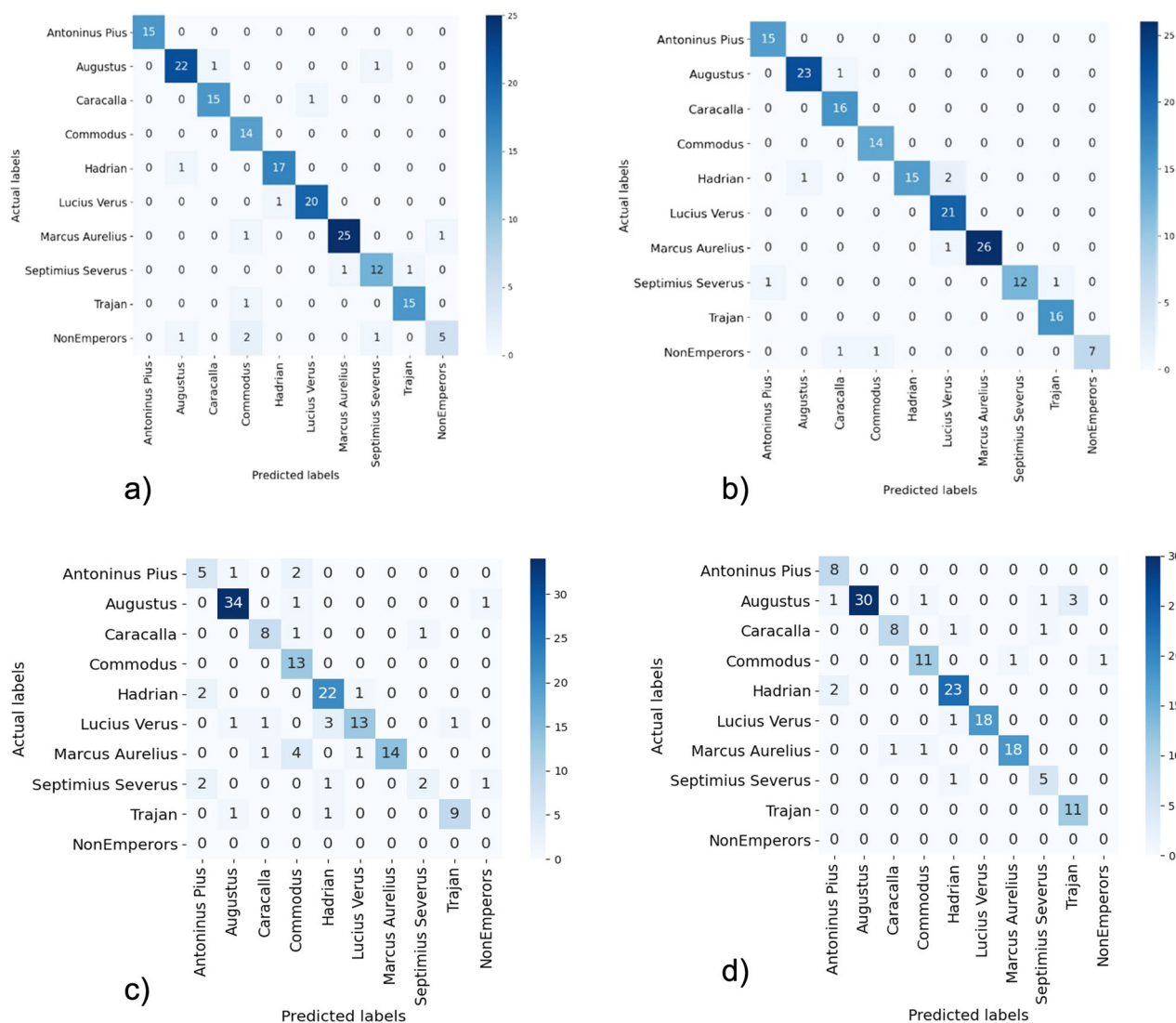


Fig. 8 Confusion matrices. **a** Experiment 1: validation set. **b** Experiment 2: validation set. **c** Experiment 1: test set. **d** Experiment 2: test set.

Table 3 F1-scores per class of validation and test set.

Emperor name	Validation			Test		
	Number of images	Exp 1 F1-score	Exp 2 F1-score	Number of images	Exp 1 F1-score	Exp 2 F1-score
Antoninus Pius	15	1.00	0.97	8	0.59	0.84
Augustus	24	0.92	0.96	36	0.93	0.91
Caracalla	16	0.94	0.94	10	0.80	0.84
Commodus	14	0.88	0.97	13	0.76	0.85
Hadrian	18	0.94	0.91	25	0.85	0.90
Lucius Verus	21	0.95	0.93	19	0.76	0.97
Marcus Aurelius	27	0.94	0.98	20	0.82	0.92
Non-emperors	9	0.67	0.88	0	0.00	0.00
Septimius Severus	14	0.86	0.92	6	0.44	0.77
Trajan	16	0.94	0.97	11	0.86	0.88
Total/weighted average	174	0.92	0.95	148	0.81	0.90

(198–217) and Geta (209–211) in copies of their second successor type, and for the emperors of the Tetrarchy (293–312). To a lesser extent, the same strategy of self-representation was used by the emperors of the Julio-Claudian and Antonine dynasties in order

to convey their dynastic ties. The misclassified image of Marcus Aurelius as his son Commodus (and vice versa) in our test may be the result of this representational strategy (Fig. 10, nos 9, 14). Similarly, the misclassification of Hadrian as Antoninus Pius may

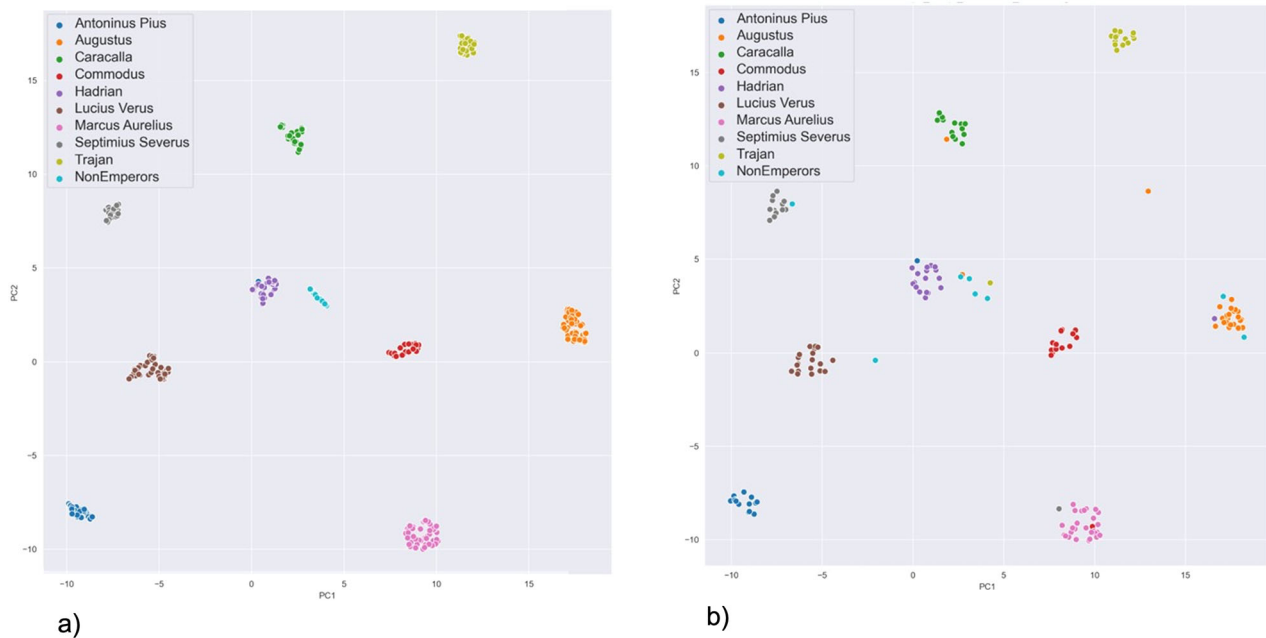


Fig. 9 UMAP projection of embeddings. **a** Embeddings of the training set. **b** Embeddings of the validation set. Each point in the maps represents an image embedding of the colour coded emperor. These are points taken from the penultimate dense layer of the model from experiment 2, which is a 512-dimension representation. Observe that the same emperor embeddings are grouped well together and are well separated from one another.

be the result of the intended close physiognomy of the portraits of Hadrian and his adopted son (Fig. 10, nos 11–12).

It is also possible that some misclassifications may be the result of the fact that the emperor's portrait often became the benchmark of a *Zeitstil* or fashion trend. This is one of the reasons why it is sometimes difficult to distinguish between a portrait of an emperor and that of a private individual.

Other misclassifications seem to be the result of a general lack of facial features due to damages to the portrait head. This is seemingly the case for three portraits of the emperor Augustus that were wrongly identified as other emperors (see Fig. 10, nos 1, 5–6). In other cases, overt modern restorations to the portraits (particularly around the nose area) seem to have send our model astray (see Fig. 10, nos 9, 10, 13). What is still worth noting is that in many cases (including the three portraits of Augustus), the correct emperor is named as the second predicted label (see Fig. 10, nos 1–5, 8–13).

Conclusion and future work

The main question this study aimed to answer is: can existing facial recognition methods based on artificial intelligence (AI) models be utilised to identify the portraits of Roman emperors? Based on our results our answer is moderately positive.

We have taken a first step towards applying facial recognition models to the study of ancient imperial portraiture. The results of our experiments have shown that by training only a few layers, a pre-existing DCNN model is able to correctly classify most images in our dataset of Roman emperors. Furthermore, the model has gained knowledge of the different emperor classes to such an extent that it is able to effectively cluster images of the same class of emperors together (as shown by the UMAP).

The performance of the model is sound, with an F1-score of 0.95 for our validation set and a score of 0.90 for our test set. Misclassifications may be the consequence both of portraits lacking facial features due to damage, and because the portraits actually being similar, both due to a phenomenon called *Bildnisangleichung*—which is well attested for some of the emperors in our dataset—and as a result of the fact that the emperor's portrait often became the

benchmark of a *Zeitstil* or fashion trend. In future research we aim to explore how FR techniques can be utilised not just to recognise different emperor classes, but also to effectively distinguish between the image of an emperor and that of a private individual.

Data availability

The dataset generated and analysed during the current study are stored on the Radboud Data Repository (<https://doi.org/10.34973/kwk9-ga38>), together with details on the experiments and the correct paths to the images used for training, validation, and testing. The dataset consists of hundreds of photographs from different sources, including personal photography, book scans, and web scrapes. The dataset is not publicly available due to copyright concerns, but is available from the corresponding author on reasonable request. Below we provide details as to how the images in our dataset were collected. The bulk of our data was collected through a web scrape (using API) from the IDAI.objects Arachne Database (<https://arachne.dainst.org/>), with permission of the director Prof. Dr. Reinhard Foertsch, who has also kindly provided us with access to high-resolution images. Search terms matched the names of the emperors (NB: in German spelling) that were analysed during the study (e.g., “Augustus”, “Hadrianus”, etc.). Furthermore, we applied the following filters to our initial search results: *Category Einzelobjekte, Contains Images ja, Sub-categorie Plastik, Object category Portrait, Subject Kaiser*. After creating the correct HTML links, we retrieved the data using API, on which see <https://arachne.dainst.org/info/apis>. The retrieved images were stored as a separate dataset. Subsequently, the data was cleaned manually (duplicates, *en profil* images, images of other emperors/non-emperors, images of modern (i.e., non Roman) portrait heads, and empty records were removed). This data was supplemented by (1) scans made from the book series *Das römische Herrscherbild*, (2) images in the archive of the Centre for Art Historical Documentation at Radboud University, and (3) images from the authors' personal collections consisting of photographs of imperial portraits from various European museums and collections. The codes used in our experiment are available at <https://github.com/Darshan-Ramesh/EmpRecognition>.

	1	Actual label: Augustus Predicted label 1: Septimius Severus Predicted label 2: Augustus		9	Actual label: Commodus Predicted label 1: Marcus Aurelius Predicted label 2: Commodus
	2	Actual label: Augustus Predicted label 1: Antoninus Pius Predicted label 2: Augustus		10	Actual label: Commodus Predicted label 1: Non emperor Predicted label 2: Commodus
	3	Actual label: Augustus Predicted label 1: Trajan Predicted label 2: Augustus		11	Actual label: Hadrian Predicted label 1: Antoninus Pius Predicted label 2: Hadrian
	4	Actual label: Augustus Predicted label 1: Trajan Predicted label 2: Augustus		12	Actual label: Hadrian Predicted label 1: Antoninus Pius Predicted label 2: Hadrian
	5	Actual label: Augustus Predicted label 1: Trajan Predicted label 2: Augustus		13	Actual label: Lucius Verus Predicted label 1: Hadrian Predicted label 2: Lucius Verus
	6	Actual label: Augustus Predicted label 1: Commodus Predicted label 2: Trajan		14	Actual label: Marcus Aurelius Predicted label 1: Commodus Predicted label 2: Trajan
	7	Actual label: Caracalla Predicted label 1: Hadrian Predicted label 2: Commodus		15	Actual label: Marcus Aurelius Predicted label 1: Caracalla Predicted label 2: Septimius Severus
	8	Actual label: Caracalla Predicted label 1: Septimius Severus Predicted label 2: Caracalla		16	Actual label: Septimius Severus Predicted label 1: Hadrian Predicted label 2: Antoninus Pius

Fig. 10 Misclassifications experiment 2: test set. Images nos 1–13, 15–16 by the Arachne object database (licensed under CC license BY-NC-ND 3.0) and no. 14 authors' own.

Received: 4 March 2021; Accepted: 9 February 2022;
Published online: 14 March 2022

Note

1 E.g., Burdur, Archaeological Museum, inv. K.95.26.08 (Marcus Aurelius).

References

Ando C (2000) *Imperial ideology and provincial loyalty in the Roman Empire*. Berkeley
 Buslaev A, Iglovikov VI, Khvedchenya E, Parinov A, Druzhinin M, Kalinin AA (2020) Albumentations: fast and flexible image augmentations. *Information* 11(2):125. <https://doi.org/10.3390/info11020125>

Dargan S, Kumar M, Ayyagari MR, Kumar G (2020) A survey of deep learning and its applications: A new paradigm to machine learning. *Arch Comput Methods Eng* 27(4):1071–1092
 Du H, Shi H, Zeng D, Mei T (2022) The Elements of End-to-end Deep Face Recognition: A Survey of Recent Advances. *ACM Computing Surveys (CSUR)*
 Fejfer J (2008) *Roman portraits in context*. Berlin
 Fittschen K (1971) Zum angeblichen Bildnis des Lucius Verus im Thermen-Museum. *Jahrbuch des Deutschen Archäologischen Instituts* 86:214–252
 Fittschen K (2010) The portraits of Roman emperors and their families: Controversial positions and unsolved problems. In: Edwald BC, Noreña C (eds.) *The emperor and Rome: Space, representation, and ritual*. Cambridge, pp. 221–246
 Guo Y, Liu Y, Oerlemans A, Lao S, Wu S, Lew MS (2016) Deep learning for visual understanding: a review. *Neurocomputing* 187:27–48

- Haines CR (1919) *Fronto*. Correspondence, vol I. Cambridge, MA, USA
- Lahusen G (2010) *Römische Bildnisse: Auftraggeber, Funktionen, Standorte*. Mainz
- LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. In: *Proceedings of the IEEE* 86(11), pp. 2278–2324
- Liu G, Reda FA, Shih KJ, Wang TC, Tao A, Catanzaro B (2018) Image inpainting for irregular holes using partial convolutions. In: *Proceedings of the European conference on computer vision (ECCV)*, pp. 85–100
- McInnes L, Healy J, Melville J (2018) Umap: uniform manifold approximation and projection for dimension reduction, <https://arxiv.org/abs/1802.03426>
- Miniae S, Boykov YY, Porikli F, Plaza AJ, Kehtarnavaz N, Terzopoulos D (2021) Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. <https://arxiv.org/abs/2001.05566>
- Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T (2019) Pytorch: An imperative style, high-performance deep learning library. In: *Advances in neural information processing systems*, pp. 8026–8037
- Pfanner M (1989) Über das Herstellen von Porträts. Ein Beitrag zu Rationalisierungsmaßnahmen und Produktionsmechanismen von Massenware im späten Hellenismus und in der römischen Kaiserzeit. *Jahrbuch des Deutschen Archäologischen Instituts* 104:157–257
- Phillips PJ, Yates AN, Ying H, Noyes E, Jackson K, Cavazos JG, Jeckeln G, Ranjan R, Sankaranarayanan S, Jun-Cheng C, Castillo CD, Chelappa R, White D, O’Toole AJ (2018) Face recognition accuracy of forensic examiners, super-recognizers, and face recognition algorithms. In: *Proceedings of the National Academy of Sciences*, pp. 6171–6176
- Pollini J (2020) Die Umarbeitung römischer Kaiserbildnisse: Deutungsprobleme und neue Lösungsansätze mit Hilfe digitaler Technologie. In: Greub T (ed.) *Revisionen des Porträts. Jenseits von Mimesis und Repräsentation*, Leiden, pp. 243–270
- Riccardi LA (2000) Unconventional imperial portraits in the eastern Roman provinces: The case of the Kanellopoulos emperor. *Hesperia* 69:105–132. <https://doi.org/10.2307/148367>
- Schofield D, Lorenz K, Davy-Jow S, Anderson M (2012) Roman portraiture and biometric identification. *Electronic Visualisation and the Arts (EVA 2012): Proceedings*. In: Dunn S, Bowen JP, Kia Ng K (eds.). Swindon: British Computer Society, Swindon. pp. 163–171
- Stewart P (2003) *Statues in Roman society: representation and response*. Oxford
- Szegedy C, Ioffe S, Vanhoucke V, Alemi A (2017), Inception-v4, Inception-ResNet and the impact of residual connections on learning. *AAAI Conference on Artificial Intelligence, North America 2017*. <https://www.aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14806/14311>. Accessed 1 Oct 2021
- Timsler (2019) Pretrained Pytorch face detection (MTCNN) and recognition (InceptionResnet) models. <https://github.com/timsler/facenet-pytorch>. Accessed 4 Mar 2021
- Wang ZJ, Turko R, Shaikh O, Park H, Das N, Hohman F, Kahng M, Chau DHP (2020) CNN explainer: learning convolutional neural networks with interactive visualization. *IEEE Trans Vis Comput Graph* 27(2):1396–1406
- Yosinski J, Clune J, Bengio Y, Lipson H (2014) How transferable are features in deep neural networks? In: *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*. MIT Press, Cambridge, MA, USA, pp. 3320–3328
- Zaidi SSA, Ansari MS, Aslam A, Kanwal N, Asghar M, Lee B (2021) A survey of modern deep learning based object detection models. <https://arxiv.org/abs/2104.11892>
- Zeiler MD, Fergus R (2014) September. Visualizing and understanding convolutional networks. In: *European conference on computer vision*. Springer, Cham, pp. 818–833
- Zhang K, Zhang Z, Li Z, Qiao Y (2016) Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Sig Process Lett* 23(10):1499–1503. <https://doi.org/10.1109/LSP.2016.2603342>
- Zhang R, Isola P, Efros AA (2016) October. Colorful image colorization. In: *European conference on computer vision*. Springer, Cham, pp. 649–666
- Zhuang F, Qi Z, Duan K, Xi D, Zhu Y, Zhu H, Xiong H, He Q (2020) A comprehensive survey on transfer learning. In: *Proceedings of the IEEE* 109(1), pp. 43–76

Acknowledgements

This research was supported by an Ammodo Humanities Award 2017 (Hekster).

Competing interests

The authors declare no competing interests.

Ethical approval

Not applicable as this study did not involve human participants.

Informed consent

Not applicable as this study did not involve human participants.

Additional information

Correspondence and requests for materials should be addressed to Darshan Srirangachar Ramesh.

Reprints and permission information is available at <http://www.nature.com/reprints>

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2022