Face Detection in Historic Manuscripts
A Case Study on the Wenceslas Bible

Heinz Hofbauer[1] • Julia Hintersteiner[2] • Manfred Kern[2] • Andreas Uhl[1]

[1]Paris Lodron University of Salzburg, Austria ({hofbauer, uhl}@cs.sbg.ac.at)
[2]Paris Lodron University of Salzburg, Austria ({julia.hintersteiner, manfred.kern}@plus.ac.at)

July 9, 2025

Abstract

We want to analyse the faces depicted in the Wenceslas Bible, an illustrated bible from the late 14th century (circa 1390), however, this requires automatic finding of faces in the illustrations of the bible. This is a difficult task due to the fact that we are working with illustrations of human faces that are often interwoven into the background and in odd poses. This paper presents an analysis of prominent face detection methods and how their performance translates from real-world facial images to painted imagery. We will make use of a scale and rotation cascade on top of these methods to see if the detection of faces in odd poses can be improved. Finally, most methods are designed to handle a particular face size, but for the purpose of finding faces in the illustration of historical books the relative size of the image and face differs from most real-world cases. An attempt to fix this is to use tiling of the input image to adjust for the relative scale difference. We will see that tiling, scaling, and rotation help, and while no general "best" setting can be given, we also see that these can boost some methods from non-working to being quite good.

Contents

1 Introduction

Detection of faces in art is relevant to many art historic questions, see Section 2 for some examples. Our reason for wanting face detection, and another example why it is of interest in an art historic context, is as follows.

This work is part of an attempt to digitize information about the Wenceslas Bible[1]. The Bible is stored in the Austrian National Library, it consists of six manuscripts with shelf marks codex 2759 to 2764. One of the tasks is to try to figure out how many people illustrated the Bible, as well as which parts each person did. There is an attribution of illustrations to specific artist [2], some of them identifiable by name, because they have signed their illustrations with initials, however, this is not an assured connection since in most cases there are no historical records about the illustrators.

Our goal is to use computer vision, taking hints from face biometry, similar to what [3] did for renaissance painters, to find commonalities in how faces are painted. The basic assumption is that recurring characters, such as Wenceslas and the "bathmaid", are painted similarly by the same painter. The painters likely never saw the king, and figures like the bathmaid are entirely fictional, we assume that an idealized version is painted, resulting in a similar outcome for the same painter. The assumption is that a similarity in face recognition indicates that the same painter created the assessed faces.

Before we can analyse the faces we have to find them first, ideally in an automated way. Face detection is a well established area of research, but it is unclear how face detection methods designed and trained on real faces will perform with painted faces. The non-realistic style of the face illustrations, with proportions being not quite right, exacerbates the detection problem. See figure 2 for a comparison of a real faces and an illustration from the Wenceslas Bible. Note that sometimes the faces are interwoven in the background or other parts of the illustration, although this is more common in the marginalia, where Wenceslas and the bathmaids typically appear.

The relatively small faces, as compared to regular photos where faces are often the focal point, and frequent slanted depiction, as a narrative device (bowed heads) or as part of the story (sleeping, dead, ...), suggest the use of a rotation and scale cascade or tiling (to fix relative face size issues).

While our evaluation is driven by our own requirements we will make general statements too, which will help others with similar problems. Our contribution to a wider audience thus are as follows. We show that single shot detectors are not a good fit for art, but an additional rotation step is beneficial. So can be a scaling or tiling approach depending on the image size and relative scale of faces to the image. We give details on how to aggregate detections from these detection steps. We show that CNNs should not be excluded from a task based on architecture alone, as the same CNN with a different training can perform vastly better or worse. This is especially important when working with few training samples of a specific art styles or time periods where pre-trained networks are used.

2 Related Work

2.1 Face Detection in Art

Not a lot of work was done on face detection in 14th century book illustrations but some on the more general topic of faces in art. Srinivasan et al. [3] used facial landmarks and paint styles to identify whether different Renaissance portraits are from the same painter. Similarly, Zhong [4] applied face biometric recognition to Song dynasty paintings as a further argument in art historical discussions, such as whether a painter had self inserted their likeness into a painting. Liang [5] performed age and gender classification of faces in Japanese art starting from cropped faces, their result was that an ensemble of different CNNs worked best. Sindel et al. [6] created ArtFacePoints, a facial landmark detector on cropped facial images in art. It takes a two-stage approach, using a coarse scale of the image to generate a rough map of the landmarks and refining them on the full resolution image. This work is a promising next step, provided faces are detected properly, to obtain stable facial landmarks for pose correction. The topic of face detection was not a focus in any of these papers. In Wechsler et al. [7], the topic of face detection is more prominent. They introduced a "faces in art" database and analysed a few face detectors. They dealt with modern art, a different topic than ours, and they highlighted that different art styles impact face detection differently. Bengamra et al. [8] performed face detection on Tenebrism style paintings using various networks. RestNet50 with a retrained Faster RCNN yielded the best results. To handle difficult poses, images in the augmented dataset were rotated by by $\pm 45°$ for retraining. In [9] they demonstrated that the retrained network can detect faces with more accuracy through the use of perturbation based explainable AI. So far, there is no systematic evaluation of face detectors for art that directly compares real-world performance and performance on painted faces. This is the goal of this work.

2.2 Face Detectors in this Work

There are a lot of face detection frameworks, see recent surveys [10, 11] for an overview, so a selection had to be made. The Deepface/Lightface framework [12] provides a good selection of face detectors which is why we chose it as a basis for this evaluation. Since RetinaFace is used a number of times in the Deepface framework, and the author shows a slight drop in performance (1-2%) in relation to the original implementation, we also included the original implementation in the evaluation. Below we will give a brief summary of the methods used and their references for further detail.

We use two versions of RetinaFace, one provided by the authors in the Insightface toolkit [13], and one implementation provided by the Deepface [12] toolkit. Both versions are based on ResNet50, however, differences in training results in a slightly different performance, see Figure 2a for a comparison. The differences are significant enough that we included both in our evaluation. The Deepface implementation will be referred to as *retinaface* and the Insightface version as *insightface*. A further version of ResNet will be used as the backbone of a single

shot detector (*SSD*) [14]. The SSD automatically generates different scales and generates multiple patches on each scale to find face candidates.

The multitask cascade convolutional network (MTCNN) [15] uses an image pyramid to perform single-stage detection. The images are then fed through a series of three networks to propose bounding boxes, perform regression on the proposed bounding boxes, and a final network that produces the output of facial position and landmark localizations. We also use the "you only look once" CNN version 8 for faces (YOLO v8). The YOLO series of CNNs are general detector networks instead of repurposed classifiers. Thus, the networks are smaller and faster, yet still produce good results. YOLO was introduced in 2016 [16] but has since seen constant development, including various offshoots. See [17] for an overview of the different YOLO CNNs and their development. The YuNET [18] is also a custom-built face detector with the goal of minimizing hardware usage and maximizing speed in a speed/accuracy trade-off. The trade-off in accuracy usually means it is less tightly targeting human faces, which might be a benefit when using it to find drawings of human faces instead of photos of human faces.

In addition, we also use traditional detectors. The used implementations are from the Deepface package [12]. For traditional methods, we utilized OpenCV's Viola Jones, which employs Haar cascades to detect faces and eyes [19], and the DLib face detector, which is based on the histogram of oriented gradients (HOG) [20]. Both methods have numerous extensions and improvements. A comparison and more in depth details can be found in [21].

Algorithm 1 Calculation of intersect in a one on one correspondence.

$G \leftarrow \mathcal{G}$
$D \leftarrow \mathcal{D}$
intersect $\leftarrow 0$
while $|G| > 0$ **and** $|D| > 0$ **do**
  $R_g, R_d \leftarrow \underset{R_g \in G,\, R_d \in D}{\arg\max} \frac{A(R_g \cap R_d)}{\max\left(A(R_g), A(R_d)\right)}$
  $D \leftarrow D \backslash \{R_d\}$
  $G \leftarrow G \backslash \{R_g\}$
  intersect $\leftarrow$ intersect $+ A(R_g \cap R_d)$
end while
return intersect

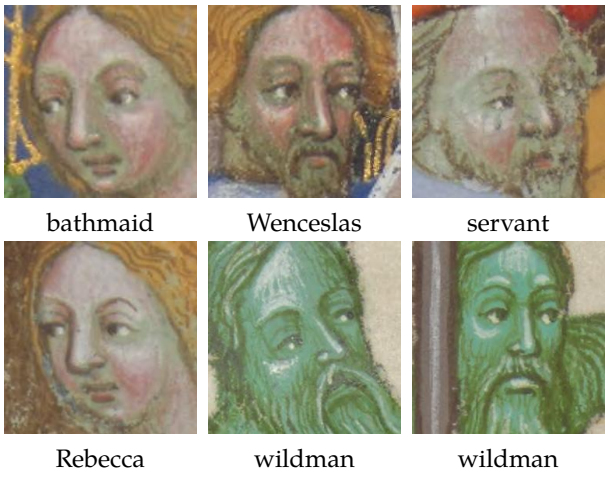


Figure 1: Ground truth samples from Genesis, Codex 2759 sheet 21 front.

3 Experiments and Discussion

We will use the F1-score to assess the accurate detection of faces, which is a well known measure from the field of information retrieval, e.g., [22], and is also typically used for face detection assessment. The F1-score is the harmonic mean of precision, i.e., facial areas that match the ground truth, and recall, i.e., the amount of the ground truth which was correctly found. Let tp be the number of correctly detected faces, fp be the number of wrongly detected faces and fn be the number of faces not detected. The precision is defined as $\mathcal{P} := \frac{\text{tp}}{\text{tp}+\text{fp}}$, and the recall as $\mathcal{R} := \frac{\text{tp}}{\text{tp}+\text{fn}}$, and finally the F1-score F1 $:= \frac{2\mathcal{P}\mathcal{R}}{\mathcal{P}+\mathcal{R}}$.

In our case the true/false positive/negative are not so straightforward, as we are not dealing with retrieved pixels but with rectangular facial areas. Specifically, two retrieved face regions can overlap a single ground truth area. When we use a pixel based precision and recall we lose the information about correspondence, but we want a single retrieved face area per ground truth faces area, i.e., a one on one correspondence. So we use the following method to gain the precision and recall values which match our purpose.

We have two sets of axis-parallel rectangular areas, one for the ground truth $\mathcal{G}$ and one for the detected faces $\mathcal{D}$ by detector f (see Section 2). We then need the correct intersection of detected vs. ground truth areas in a one on one correspondence, denoted as intersect, with $A(r)$ denoting the area of rectangle r as given in algorithm 1. We can then calculate our precision and recall with $\mathcal{P} := \frac{\text{intersect}}{\sum_{R \in \mathcal{D}} A(R)}$, and $\mathcal{R} := \frac{\text{intersect}}{\sum_{R \in \mathcal{G}} A(R)}$.

The algorithms we use all follow this basic idea: a single stage detects a face only once. To combine multiple detections there must be a one on one correspondence, consequently, we cannot combine a face representation more than once. This is reflected in the set reduction part of the algorithms. To prevent small overlaps of adjacent faces from being counted as referencing the same face, we work from the largest match in an ordered fashion. To prevent overlapping faces in crowd situations from being counted as belonging to the same face we only count overlapping rectangles as indicating the same face when they are either a) overlapping with at least 50% area (calculated on the largest rectangle) or b) one is fully contained in the other and the area is at least 20% of the larger area. For an evaluation of how overlap affects performance, although for modern art, refer to [7].

For the evaluation we need a ground truth, thus we manually segmented Genesis from the Wenceslas Bible with axis-parallel bounding boxes. We segmented all human and human-like, e.g., monkeys and wildmen, faces. Some faces, in illustrations depicting crowds, are quite obfuscated, so we only used faces showing two of the following four features: left eye, right eye, mouth, nose. This resulted in 415 faces on 50 pages, some examples can be seen in Figure 1.

3.1 Evaluation of face detectors

A common comparison of our selected face detection methods is not known to us. So we will perform an evaluation and comparison between methods and source material, i.e., real and painted faces. For real faces we use the face detection data set (FDDB) [23], containing 5171 faces in 2845 images with corresponding ground truth and for art faces we use the manual segmentation of the Genesis chapter of the Wenceslas Bible as described above. Average F1-scores over the database are given per method. For art faces, due to the large image size and textured page, many small details are detected as faces. We drop any rectangle which is not at least 75×75 pixels in size. We calculated the F1-score per page, then averaged them for the final score per detector. Scores for the evaluation are given in Table 1 and sample detections are given in Figure 2, note that we used a crowd image for real faces rather than an image from the FDDB to better showcase the detection performance of the algorithms.

For real faces the CNN-based methods outperform the traditional methods (DLib and OpenCV), YuNet, being built for speed sacrifices accuracy. The *retinaface* implementation is about 2% below the *insightface* implementation as stated by its author. Only the SSD, which also has a ResNet50 as a backbone, like RetinaFace, exhibits a poor performance when compared to *retinaface* and *insightface*, which is also reflected in the sample image.

For art faces the performance is very different. The YuNet sacrifices accuracy for speed, so the assumption was that it is not as well adapted to specifically realistic human faces and would generalize better, a faulty assumption. The RetinaFace based methods are interesting in that they perform so differently, *insightface*, while slightly worse compared to real faces, performs well, *retinaface*, which was comparable to *insightface* on regular images, has a large drop in performance, below even the traditional methods, and SSD fails to detect any face. This is likely a combination of training data and scaling assumptions, suggesting that refinement training is useful, or in some cases required. The decision how to handle different scale can have a huge impact, which can be seen in the different performance drops, when compared to the FDDB F1-scores, of MTCNN (≈ -0.25), YOLOv8 (≈ -0.75) and SSD (a drop to 0). The commonality here seems to be that the pyramidal CNNs are outperforming the one-step networks (YOLOv8 and SSD). The architectures of MTCNN (cascade networks) and RetinaFace (single stage) are different, but both use a pyramidal approach, the MTCNN to generate face candidates and RetinaFace uses a feature pyramid. But YuNet also utilizes a feature pyramid (tiny feature pyramid network) so that alone cannot be the only explanation.

3.2 Combining Tiling, Scaling and Rotations

While we have too little data with ground truth to perform refinement training at this stage we can implement a scaling and rotation cascade on top of the detector methods. The reason to include rotation is that people look down or up, as a narrative device, and people lying down, dying, sleeping and fornicating, are surprisingly frequent, cf. Figure 2b. We include a scaling step because the scale of the page and the size of the faces in the pages are outside the typical scenarios of real images. Also, face size varies depending on how prominent the face is in a given illustration. The small relative size of faces to the overall page, can also be fixed by tiling, i.e. overlapping crops of the page. If we use a crop of the page the relative size of the face is returned to (roughly) in the same relation as in real photos. The tiles we use are square and of size 3000, which is similar to digital photos which are typically in the 4000×3000 range, while pages from the Wenceslas Bible are three to four times that size. As an example the ratio of face to image height for person focused images (like our FDDB) is around $1:4$ and around $1:20$ for regular images containing people, like the large crowd image (Fig. 2a). The ratio in the Wenceslas Bible is around $1:122$ which is reduced to $1:25$ with tiling. We use a 50% overlap of tiles to prevent the non-detection of faces which otherwise would be cut in half on the border of the tile.

In essence we will build a multistage detector since the single shot detectors we tested do not work. However, we don't blindly test multiple options and optimize classification like with test-time augmentation [24], rather we investigate multiple options in a targeted manner to get an optimal set of operations for detection, and also runtime cost, by minimizing the number of stages required. When a face is found on multiple scales, rotations or tiles it will have more than one rectangle representing it, this has to be solved. The solution is quite easy: either average the rectangles to get a more stable representation or to use the rectangle with the highest confidence score to get the best localization. We chose the first, by averaging left, right, top and bottom coordinates of the rectangle respectively, to allow for methods which do not provide confidence scores.

The results of the test are given in Table 2 using a scale (S) and rotation (R) cascade as given. Once without tiling (Table 2a) and once with tiling (Table 2b). The F1-scores, with the best score per detector marked in bold, are given for each possible combination of rotations and scales to get a sense for how a given scale and/or rotation impacts the results. Note that S1 in Table 2a is the same as art faces in Table 1. The results for SSD and YuNet are not included since they reflected the results in Table 1, i.e., they did not work at all. We kept the scaling and rotation relatively simple, $\pm 45°$ and $\pm 90°$ for rotation and down scaling by 2 and 4 (and unscaled 1).

Let us first look at the non-tiling cascade (Table 2a). Except for the Viola-Jones based OpenCV, which has downsampling built in by the Haar cascade, all detectors benefit from scaling, however, the gain is mostly small. Interestingly *retinaface* and *insightface* benefit differently from scale/rotation even though they have the same architecture, which means that this is a matter of training rather than architecture.

If we add tiling (Table 2b), it is of interest whether scaling is still helpful even if the method-inherent handling of scales works due the relative scale being fixed. Two things are of note: 1) as suspected additional scaling steps are no longer necessary. And 2) tiling does little for methods which already performed well on the untiled version, but it improves methods which had problems before (*retinaface* and YOLOv8). Tiling seems to be better mostly for methods which had serious problems before,

Table 1: Baseline results for the face detectors on real faces (FDDB) and art faces (Wenceslas Bible) given as F1-score.

faces	DLib	*insightface*	MTCNN	OpenCV	*retinaface*	SSD	YOLOv8	YuNet
real	0.686	0.840	0.805	0.659	0.821	0.789	0.832	0.789
art	0.521	0.798	0.586	0.287	0.264	0.000	0.062	0.000

Table 2: F1 scores for rotation, scale and tiling experiments. Highest value per detector bolded.

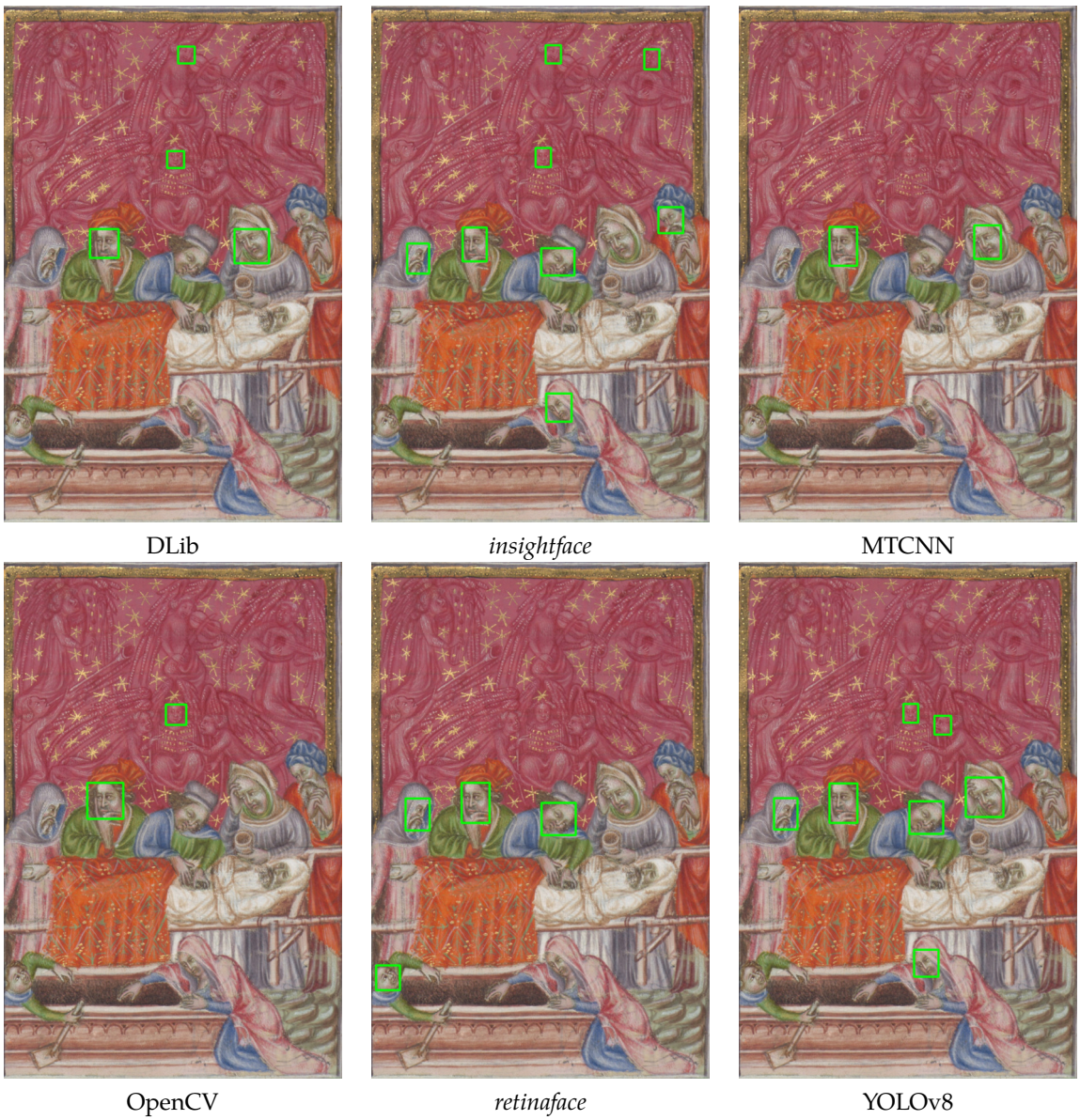
(a) Rotation and scaling without tiling.

scale and rotation	F1-score					
	DLib	*insightface*	MTCNN	OpenCV	*retinaface*	YOLOv8
S1	0.521	0.798	0.586	**0.287**	0.264	0.062
S1,2	**0.529**	0.800	**0.586**	0.267	0.292	0.066
S1,2,4	0.514	0.794	0.581	0.257	0.322	**0.082**
S1,2,4 R45	0.435	0.765	0.337	0.156	0.282	0.061
S1,2,4 R45,90	0.323	0.708	0.280	0.101	0.296	0.053
S1,2,4 R90	0.375	0.735	0.470	0.148	**0.335**	0.071
S1,2 R45	0.476	0.790	0.397	0.179	0.246	0.048
S1,2 R45,90	0.370	0.743	0.337	0.113	0.246	0.049
S1,2 R90	0.402	0.760	0.511	0.154	0.292	0.066
S1 R45	0.488	**0.804**	0.483	0.206	0.221	0.046
S1 R45,90	0.415	0.764	0.426	0.134	0.221	0.046
S1 R90	0.443	0.768	0.552	0.177	0.264	0.060

(b) Rotation and scaling with tiling.

scale and rotation	F1-score					
	DLib	*insightface*	MTCNN	OpenCV	*retinaface*	YOLOv8
S1	**0.508**	0.796	**0.585**	**0.265**	0.634	**0.595**
S1,2	0.405	0.649	0.474	0.207	0.566	0.479
S1,2,4	0.386	0.634	0.469	0.197	0.584	0.466
S1,2,4R45	0.235	0.609	0.195	0.129	0.649	0.243
S1,2,4R45,90	0.178	0.537	0.163	0.079	0.653	0.170
S1,2,4R90	0.254	0.554	0.371	0.101	0.599	0.288
S1,2R45	0.247	0.643	0.222	0.162	0.635	0.252
S1,2R45,90	0.191	0.570	0.183	0.095	0.638	0.202
S1,2R90	0.280	0.582	0.383	0.105	0.581	0.332
S1R45	0.337	**0.815**	0.378	0.244	0.724	0.358
S1R45,90	0.264	0.775	0.318	0.145	**0.732**	0.273
S1R90	0.369	0.772	0.502	0.145	0.654	0.454

(a) Samples of a real faces on the large crowd image.



(b) Art face samples on a crop of sheet 53r, codex 2759, Wencelsas Bible. SSD and YuNet are not included, they did not detect any faces.

Figure 2: Samples of face detection in art faces and real faces.

but *insightface* also improves from 0.804 to 0.815 F1-score with tiling. For any given method it is not known a-priori what the cause of the problem is, i.e., is the problem internal scale handling which cannot deal with the large relative scales or is the problem the inability to detect painted faces. Therefor, an evaluation is necessary for any given method, i.e., the findings can not be generalized, but testing on some cropped images seems to be sufficient.

Whether a rotation step is beneficial again strongly depends on the method but seems to be independent of scaling. Tiling can help by reducing the number of false positives, which allows *retinaface* to use two rotation steps with tiling vs. one without. However, it is also training related, as can be seen by the difference between *insightface* and *retinaface* which otherwise have the same architectures.

A further note on the generalization of these results. Not shown in Table 2 are the components of the F1-score which are based on finding all faces and miss-detection of faces. Rotation and scale in all cases improved the detection rate but also the miss-detection rate. A problem with rotation is that while there are rotated faces, they are relatively few in number, and while they are now properly detected the miss-detection rate is also increased, but that can happen anywhere. Thus the F1-score does not improve. For works where there is a higher rate of rotated faces the result may well be different.

Also note that tiling, rotation and scaling only optimize performance, in no case did they boost detection performance in a major way. That is, it is best to start with a well performing method. Unfortunately, the performance on real-world imagery does not determine the performance on painted faces, although the performance on painted faces only ever dropped in relation. Also, from the differences in *retinaface* and *insightface*, same architecture but different training, it is clear that refinement training should be performed if possible.

4 Conclusion

The performance on real world-images generally does not transfer over to painted images. An evaluation on the target imagery is required to find methods fit for use. The problems are not based on architecture, the original implementation of Retinaface (*insightface* in the paper) and the reimplementation, Retinaface by lightface/deepface (*retinaface* in the paper), show this. While *insightface* performed well, *retinaface* does not, and that is despite a comparable performance of both methods on real-world images. In our evaluation the original Retinaface implementation [13] was the best performing method for painted faces.

We have seen that the relative size of faces to the size of the page can affect detection rates. Both *retinaface* and YOLOv8 suffer from this, although they otherwise show good performance on painted faces, as does MTCNN. A tiling approach, in which the relative size of the faces in the image is closer to regular photography, helps to address this problem. But, tiling does not improve matters for all cases, it must be evaluated for each algorithm.

We have shown that rotation or scaling improves the performance of most methods, though combining scaling and tiling does not improve the performance. The exact settings, again, differ for each method, so no general advice can be given. For general use it seems the best method is *insightface* with tiling and a rotation of $\pm 45°$.

Acknowledgment

This project received funding from the Salzburg State Digital Humanities project "Digitalisation in the Humanities, Social and Cultural Sciences (HSC)" [1].

References

[1] E. K. des Fachbereichs Germanistik der Universität Salzburg und der Österreichischen Nationalbibliothek. "Die Wenzelsbibel − Digitale Edition und Analyse." version 2.2.0. Webpage: https://edition.onb.ac.at/wenzelsbibel. (2024), (visited on 01/25/2024) (cit. on pp. 2, 7).

[2] M. Theisen and U. Jenni, *Mitteleuropäische Schulen IV (ca. 1380–1400); Textband: Hofwerkstätten König Wenzels IV. und deren Umkreis*. 2014. DOI: 10.26530/oapen_507994 (cit. on p. 2).

[3] R. Srinivasan, C. Rudolph, and A. K. Roy-Chowdhury, "Computerized face recognition in renaissance portrait art: A quantitative measure for identifying uncertain subjects in ancient portraits," *IEEE Signal Processing Magazine*, vol. 32, no. 4, 2015. DOI: 10.1109/MSP.2015.2410783 (cit. on p. 2).

[4] G. Zhong, "A computer vision-aided analysis of facial similarities in song dynasty imperial portraits," *Electronic Imaging*, vol. 35, no. 13, 2023. DOI: 10.2352/EI.2023.35.13.CVAA-212 (cit. on p. 2).

[5] Z. Liang, "Face recognition from art face images based on deep learning," in *Proceedings of the 4th International Conference on Big Data Research*, ser. ICBDR '20, 2021, ISBN: 9781450387750. DOI: 10.1145/3445945.3445963 (cit. on p. 2).

[6] A. Sindel, A. Maier, and V. Christlein, "Artfacepoints: High-resolution facial landmark detection in paintings and prints," in *Computer Vision – ECCV 2022 Workshops*, 2023. DOI: 10.1007/978-3-031-25056-9_20 (cit. on p. 2).

[7] H. Wechsler and A. S. Toor, "Modern art challenges face detection," *Pattern Recognition Letters*, vol. 126, 2019, ISSN: 0167-8655. DOI: 10.1016/j.patrec.2018.02.014 (cit. on pp. 2, 3).

[8] S. Bengamra, O. Mzoughi, A. Bigand, and E. Zagrouba, "New challenges of face detection in paintings based on deep learning," in *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications-Volume 4: VISAPP*, 2021 (cit. on p. 2).

[9] S. Bengamra, O. Mzoughi, A. Bigand, and E. Zagrouba, "Towards explainability in using deep learning for face detection in paintings," in *Proceedings of the 12th International Conference on Pattern Recognition Applications and Methods - Volume 1: ICPRAM,*, 2023. DOI: 10.5220/0011670300003411 (cit. on p. 2).

[10] S. Minaee, P. Luo, Z. Lin, and K. Bowyer, "Going deeper into face detection: A survey," *arXiv preprint arXiv:2103.14983*, 2021. DOI: 10.48550/arXiv.2103.14983 (cit. on p. 2).

[11] Y. Feng, S. Yu, H. Peng, Y.-R. Li, and J. Zhang, "Detect faces efficiently: A survey and evaluations," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 4, no. 1, 2022, ISSN: 2637-6407. DOI: 10.1109/tbiom.2021.3120412 (cit. on p. 2).

[12] S. I. Serengil and A. Ozpinar, "Hyperextended lightface: A facial attribute analysis framework," in *2021 International Conference on Engineering and Emerging Technologies (ICEET)*, IEEE, 2021. DOI: 10.1109/ICEET53442.2021.9659697 (cit. on pp. 2, 3).

[13] J. Deng, J. Guo, E. Ververas, I. Kotsia, and S. Zafeiriou, "Retinaface: Single-shot multi-level face localisation in the wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020 (cit. on pp. 2, 7).

[14] W. Liu *et al.*, "Ssd: Single shot multibox detector," in *Computer Vision – ECCV 2016*, 2016. DOI: 10.1007/978-3-319-46448-0_2 (cit. on p. 3).

[15] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, 2016, ISSN: 1070-9908. DOI: 10.1109/LSP.2016.2603342 (cit. on p. 3).

[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016 (cit. on p. 3).

[17] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, "A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas," *Machine Learning and Knowledge Extraction*, vol. 5, no. 4, 2023, ISSN: 2504-4990. DOI: 10.3390/make5040083 (cit. on p. 3).

[18] W. Wu, H. Peng, and S. Yu, "Yunet: A tiny millisecond-level face detector," *Machine Intelligence Research*, 2023. DOI: 10.1007/s11633-023-1423-y (cit. on p. 3).

[19] P. Viola and M. Jones, "Robust Real-time Object detection," in *International Journal of Computer Vision*, vol. 57, 2001 (cit. on p. 3).

[20] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (CVPR'05)*, vol. 1, 2005 (cit. on p. 3).

[21] A. Adouani, W. M. Ben Henia, and Z. Lachiri, "Comparison of haar-like, hog and lbp approaches for face detection in video sequences," in *2019 16th International Multi-Conference on Systems, Signals & Devices (SSD)*, 2019. DOI: 10.1109/SSD.2019.8893214 (cit. on p. 3).

[22] C. J. van Rijsbergen, *Information Retrieval*, second edition. 1979 (cit. on p. 3).

[23] V. Jain and E. Learned-Miller, "Fddb: A benchmark for face detection in unconstrained settings," University of Massachusetts, Amherst, Tech. Rep. UM-CS-2010-009, 2010 (cit. on p. 4).

[24] M. Kimura, "Understanding test-time augmentation," in *Neural Information Processing*, 2021, ISBN: 978-3-030-92185-9. DOI: 10.1007/978-3-030-92185-9_46 (cit. on p. 4).