A Deep Learning-Based Method for Facial Wrinkle Detection

Mohammad-Javad Rezaei, Mozafar Bag-Mohammadi, Mojtaba Karami, and Wanqing Tu

***Abstract*—In recent years, automated wrinkle detection using deep learning approaches has become prevailed due to its potential applications in fields such as skincare, cosmetics, and fashion. This paper proposes the use of Fast R-CNN and YOLOv8 for wrinkle detection in facial images. We have generated a new dataset consisting of 1022 images of human faces collected through the web with 6918 annotations made by an expert. The dataset includes varying degrees of wrinkles and was balanced for ease of training. The size of the training dataset is artificially increased from 714 to 2138 by data augmentation technique creating modified copies of existing images. The annotations for the images were classified into nine different categories based on the wrinkle location on the face. The performance of the proposed method was evaluated on a validation set and a separate test set. The mAP score was 92% for YOLOv8, 62% for Fast R-CNN with resnet-50v2. Both the wrinkle detection dataset and related codes are publicly available2, which are expected to be valuable resources for further research on AI-powered wrinkle detection.**

***Index Terms*—Facial wrinkle detection, deep learning, Fast R-CNN ResNet-50, YOLOv8, object detection, machine learning.**

I. Introduction

THE facial wrinkle assessment is important for age prediction and determination of the filler material injection type. The detection of wrinkles in facial images by computer vision techniques has increased significant attention in recent years due to its potential applications in fields such as skincare, cosmetics, and fashion. Traditional methods for wrinkle detection rely on manual inspection by trained experts, which is time-consuming and subjective. Therefore, automated wrinkle detection methods using deep learning-based methods have harvested increasing interest in recent years [1].

Deep learning algorithms have shown promising results in various computer vision tasks, including object detection, recognition, and segmentation. Among deep learning algorithms, Yolo [2] and Fast R-CNN [3] are popular choices for object detection tasks due to their high accuracy and efficiency. Fast R-CNN ResNet-50 is a variant of the Faster R-CNN algorithm [4], which consists of a region proposal network (RPN) and a Fast R-CNN network. The RPN is responsible for proposing regions of interest (ROIs) in the image that are likely to contain objects of interest. The Fast R-CNN network takes the ROIs proposed by the RPN and classifies them into different classes.

M. Bag-Mohammadi and M. Karami are with the Department of Computer Engineering, Ilam University, Ilam, Iran.

M. J. Rezaei was with Azad University, Kermanshah, Iran.

W. Tu is with Durham University, Durham, United Kingdom

Previous studies have used Fast R-CNN ResNet-50 for wrinkle detection in images with promising results. For instance, Ismail et al. [5] utilized Fast R-CNN ResNet-50 to classify and detect acne lesions and wrinkles. They achieved a 47.96% mean average precision score on a dataset consisting of 250 images. In another study, Chang et al. [6] suggested employing ResNet models for the classification of speckles and wrinkles. The classification was evaluated through K-fold cross-validation on a dataset comprising 200 images, and the results indicated that the ResNet model demonstrated superior performance in terms of average classification accuracy with more layers.

However, most of these studies have utilized small datasets with limited variations in wrinkle types and degrees of severity. Therefore, there is a need for further research using larger and more diverse datasets to validate the effectiveness of the proposed method. In this paper, we propose the use of Fast R-CNN ResNet-50 and YOLOv8 for wrinkle detection in facial images. We made a new dataset consisting of facial images with varying degrees of wrinkles. The facial images are collected from the web and annotated by an expert. The performance of the proposed method was evaluated on a validation set and a separate test set, and the results were compared with previous studies. In particular, we have made following contributions:

- We have created a new dataset for wrinkle detection applications which is publicly available. It contains 714 facial images that augmented by an expert. It contains nine different facial wrinkle classes. It allows researchers to evaluate their learning approaches accurately, act as a foundation for their future studies, and give credibility to their research.
- We have shown that the latest version of Yolo (i.e. YOLOv8) is a capable method in detecting facial wrinkle in comparison to Fast R-CNN with ResNet-50. The accuracy of YOLOv8 is 92% which outperforms previous approaches.
- The proposed method can detect objects in real time with high accuracy and efficiency, making it suitable for a real time facial wrinkle detection on live videos.

The rest of this paper is organized as follow. We review the related work in section II. The generated dataset and the proposed method are described in sections III and IV respectively. Section V is devoted to comparison and simulation results. Finally, we conclude the paper in section VI.

II. Previous Work

In recent years, there has been a significant amount of research focused on wrinkle detection in facial images. Various techniques have been proposed, including Hessian line tracking (HLT), which was introduced by Ng et al. [7]. HLT was found to improve the accuracy of wrinkle localization compared to other methods and achieved an accuracy of 84%. Canak et al. [8] utilized local binary pattern and Gabor filter techniques to automatically detect and quantify forehead wrinkles in facial images, with the Gabor filter achieving a higher accuracy of 79.7%.

Zaghbani et al. [9] proposed a novel method for facial expression recognition using Gabor wavelets to detect wrinkles in the forehead and glabella areas, achieving an average precision rate of 81.96% for recognizing seven basic emotions. Xie et al. [10] presented a novel algorithm for the automatic detection of transient wrinkles, which includes edge pair matching, wrinkle structure location, and wrinkle classification. This proposed wrinkle detector was found to be competitive in detecting different transient wrinkles and can improve the realism of expression synthesis.

Elbashir et al. [11] evaluated wrinkle detection algorithms on the whole face and proposed an enhancement technique that outperformed existing methods, achieving an average Jaccard similarity index of 56.17%. Other studies have also used algorithms for detecting nasolabial wrinkle lines using the Active Appearance Model and Hessian filter [12].

Tarasova et al. [13] developed a web application to recognize faces in photos and identify wrinkles on the face using Python, OpenCV, Dlib, and the Django framework. They utilized 68 facial landmarks to accurately detect facial features, such as eyes, nose, and mouth corners. They then used cloud computing to analyse the location and severity of various types of wrinkles on the face. The purpose of their work was to create a software system for face recognition in photos and provide a method for determining the location and severity of wrinkles on the face.

Zheng et al. [14] proposed the potential applications of automatic facial skin condition assessment, which include detecting underlying health problems and recommending skincare products. Selfies in the wild are a useful data resource for this purpose, but accurate detection of skin features is crucial. Their paper presents an automatic facial skin feature detection method that works across various skin tones, ages, and lighting conditions, which can accurately detect acne, pigmentation, and wrinkles using a two-phase annotation scheme and Unet++ network architecture.

Carlos et al. [15] proposed the importance of cutaneous relief analysis in skincare and dermatology. A new algorithm is proposed for detecting wrinkles and quantifying skin roughness using dermatoscopy. A clinical study with 33 participants showed the algorithm's comparable sensitivity to the established PRIMOS system, making it a promising alternative for evaluating dermo-cosmetic treatments.

Kim et al. [16] discuss the importance of measuring facial wrinkles as a sign of aging. Traditional image processing methods for wrinkle detection were inadequate due to the varied characteristics of wrinkles. Recently, deep learning techniques have been used, but accurate labeling of wrinkles remains challenging. In their paper, Kim et al. [16] propose a semiautomatic labeling strategy incorporating a texture map and a deep learning model, resulting in superior performance compared to existing methods when evaluated with real facial images from skin diagnosis devices.

Recent studies have also focused on deep learning-based approaches for wrinkle detection, such as Fast R-CNN ResNet-50 and ResNet models, as utilized by Ismail et al. [5] and Chang et al. [6], respectively. However, the evaluation in both studies was limited by the small size of their datasets. Therefore, more extensive evaluations on larger and more diverse datasets are needed to fully assess the performance of these deep-learning models for wrinkle detection. In another study, Deepa et al [17] present a deep convolutional neural network (CNN) to detect wrinkles on human skin, which are a common sign of aging. The proposed method is focused on using facial images to predict age and detect wrinkles, using AI, deep learning, and CNN techniques for fast performance. The method involves identifying wrinkles in a region of interest (ROI) using pattern recognition algorithms and classifiers, allowing for efficient diagnosis of skin illnesses and analysis of wrinkle stages.

Sameera et al. [18] also present the potential of deep learning and convolutional neural networks (CNN) in the cosmetic and dermatology industries, particularly for detecting premature aging. The author describes a project focused on building a CNN-based model capable of detecting features such as wrinkles, acne, and blemishes. The model achieved an overall accuracy of 94.11% and is computationally efficient compared to previous models, with diverse applications. In [19], an optimized version of Nelder-Mead algorithm is used to propose a deep learning method for identifying facial wrinkles.

Overall, previous research has shown the effectiveness of various image processing techniques and deep learning-based methods for wrinkle detection and analysis in facial images. However, there is still room for improvement in terms of accuracy and efficiency, and further research is necessary to explore the full potential of deep learning-based methods for wrinkle detection.

III. Dataset

According to recent survey on wrinkle detection [1], five major datasets have been used in literature for facial age estimation [20], facial wrinkle detection [1], and effect of smoking on facial wrinkles [21]: Bosphorus [22], FGNET [23], MORPH [24], FERET [25], and PAL [26]. Bosphorus is mainly used for facial expressions analysis using 3-dimensional (3D) modality in a controlled environment. FGNET and FERET are low resolution datasets [20] and useful for the face age estimation algorithms. MORPH is origninaly purposed for face biometrics applications. PAL is introduced to address the ineficiency of aforementioned datasets in representing the whole lifespan of face images. This dataset is also suitable for facial age estimation studies.

However, these datasets suffer from inherent limitations such as low resolution and lack of diversity in terms of

Fig. 1. Dataset samples with annotations

racial representation which make them unsuitable for effective wrinkle detection. Racial diversity is a crucial factor in accurately identifying and analyzing wrinkles. In light of these limitations, we have carefully collected our dataset from various reputable online sources, ensuring high-quality data encompassing a wide range of racial backgrounds. Also, we have captured nm photos using a high resolution camera phone (i.e. sony xxyy) using local individuals.

Our dataset contains 1022 images of human faces. These images are annotated by an expert resulting in total 6918 annotations. To annotate the data, we have used the labelImg [27] image annotation tool (by MIT, Massachusetts Institute of Technology). The annotations were classified into nine different categories based on the location of the wrinkles on the face. These categories were 1-Forehead, 2-Frown line, 3-Crow's feet, 4-Bunny line, 5-Gummy smile, 6-Smoker lines, 7-Masseter, 8-Sad smile, and 9-Chin as shown in Fig. 1. Some examples of annotated images are shown in Fig. 2. This step is critical for the dataset since the detection algorithms are usually developed using supervised deep learning. The more accurately the training data set is prepared by annotated images, the higher is the accuracy of the whole wrinkle detection and classification system.

A. Dataset Statistics

The class imbalance of generated categories is shown in Fig. 4. We observe that wrinkles on the forehead have the highest frequency among the collected data, while the bunny line has the lowest wrinkle frequency. The average image size and the median image ratio were 0.5 megapixels and 750 * 640 respectively. Dimension insights and annotation heat map of our dataset are depicted in Fig. 5. The minimum and maximum image heights are 236 and 1055 pixels. The aspect ratio of the generated images ranges from 0.4 to 2.6. The histogram of object count is shown in Fig. 6. We can see that 80% of images have between 2 and 10 annotations. Only a few images is having more than 30 annotations.

B. Dataset Augmentation

Data augmentation techniques are critical for the success of any object detection project. We utilized mosaic augmentation to increase the size of our training data as shown in Fig. 8. In

Fig. 2. Dataset samples with annotations

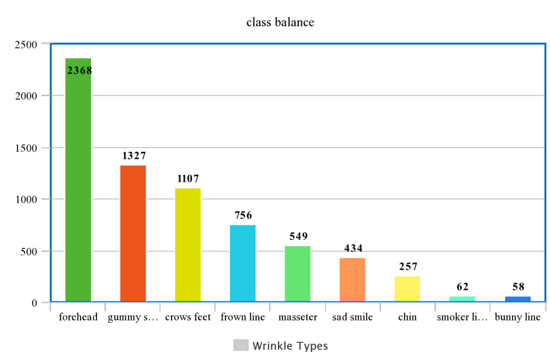

Fig. 3. The class imbalance of the collected dataset.

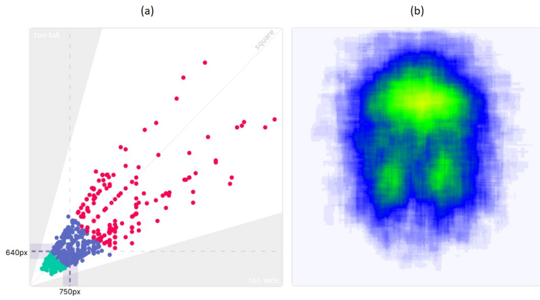

Fig. 4. (a) Dimension insight (b) annotation heat map

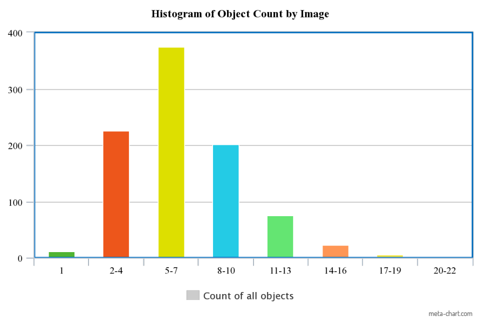

Fig. 5. Histogram of object count

mosaic augmentation, four images are stitched together with different crop ratios. At the end of this process, we had the data set contains about 2138 images, and it is ready to be forwarded to the main module (object detection and classification).

IV. Proposed Methods

A. Deep learning

Deep learning algorithms can automatically learn to recognize patterns in large datasets, such as images, speech, and text, without being explicitly programmed. This enables deep learning models to perform a wide range of tasks, including image and speech recognition, natural language processing, and even playing games at a superhuman level. Sophisticated image classification models consist of a multitude of parameters. Moreover, initiating the training process of convolutional neural networks from the beginning necessitates a labelled set of training data and an immense amount of computational power. Nevertheless, there exists a technique known as transfer learning [28] that can circumvent many of these challenges by utilizing a model that has previously undergone training on similar data and tasks, and then re-purposing it for a new model. We have decided to use YOLOv8 [2] and Fast R-CNN [4] with ResNet50v2 for our purpose.

Fig. 6. Mosaic augmentation

YOLOv8 is the latest version of the popular real-time object detection system, YOLO (You Only Look Once). Developed by Joseph Redmon and his team, YOLOv8 builds upon previous versions of YOLO by improving the accuracy and speed of object detection. YOLOv8 uses a backbone network that is based on the CSPDarknet53 architecture, which includes many layers of convolutional neural networks. It also employs a feature pyramid network to detect objects of different scales, which improves its ability to detect small objects. Additionally, YOLOv8 uses a dynamic anchor assignment method to better match object sizes, which further improves its accuracy.

Fast R-CNN (Fast Region-based Convolutional Neural Network) is a popular object detection algorithm that was developed by Microsoft Research. Fast R-CNN has shown impressive performance on several benchmark object detection datasets and has become a popular choice for object detection in many real-world applications. Fast R-CNN works by first generating a set of region proposals using a selective search algorithm, which identifies potential object locations in an image. These proposals are then passed through a deep convolutional neural network (CNN) to extract features. The extracted features are then used to classify the object and refine the bounding box coordinates of the proposed region. The key innovation of Fast R-CNN is that it performs the region proposals and feature extraction steps together in a single forward pass through the neural network, which significantly speeds up the overall processing time compared to R-CNN. Additionally, it uses a RoI (Region of Interest) pooling layer to allow variable-sized regions of interest to be mapped to a fixed-sized feature map, which further improves efficiency.

B. Methodology

The true potential of transfer learning is that the lower layers of the model, having been trained to identify specific objects, can be repurposed for a multitude of classification tasks without requiring any significant modifications. The outcomes obtained were employed to modify the weights of the proposed models throughout the training process. In this research project, we train a deep-learning image classifier to recognize nine distinct categories of wrinkles. To accomplish this, the feature extraction abilities of two robust image classifiers, namely ResNet50 v2 and YOLOv8s, were leveraged, but with a modified classification layer on top (as shown in Fig. 7).

In the initial stage, all images in each folder are analysed for every class, and then their corresponding bottleneck values are calculated and stored. The term 'bottleneck' is informally used to refer to the layer responsible for classification just before the final output layer. The learning process takes place 150 training

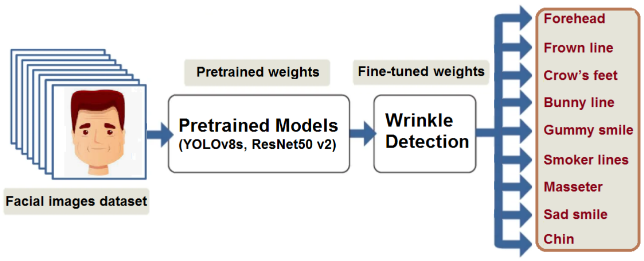

Fig. 7. Transfer Laerning

steps. For each step, Fast R-CNN uses four randomly selected images from the dataset, while YOLOv8 uses 16 images. The bottlenecks of these images are then retrieved from the cache and fed into the final layer for prediction.

To construct a model that detects wrinkles, a dataset consisting of 2138 face images featuring various types of wrinkles was compiled using YOLOv8 and Fast R-CNN. The key component of the Fast R-CNN framework is ResNet. To develop a Fast R-CNN model that can accurately detect wrinkles, the dataset was formatted in Pascal VOC format and fed into the model. The model then leveraged the power of ResNet50 v2 to identify wrinkles in the images. The central element of the YOLOv8 framework is CSPDarknet53. The dataset was formatted in a manner compatible with the YOLO model and subsequently fed into YOLOv8 for wrinkle detection.

V. Experiments and Results

To perform the experiments, we split the dataset randomly into three sets: training, validation, and test sets. The sizes of these sets are approximately 70%, 20%, and 10% of the original size of dataset. More Specifically, the training set consists of 2138 images. The validation set contains 205 images, and the test set contains 103 images. To assess the efficacy of the proposed wrinkle detection method, we compared it against state-of-the-art work by Ismail et al [5], Fast R-CNN resnet50 v2 (abbreviated as Fastv2 for the sake of brevity), and YOLOv8, with the same wrinkle initialization strategy.

A. Evaluation Metrics

The mean Average Precision (mAP) is a well-known performance metric that is frequently used to evaluate machine learning models that focus on object detection tasks and information retrieval on images, specifically using R-CNN and YOLO models [29]. mAP leverages other sub-metrics such as confusion matrix, Intersection over Union (IoU), precision and recall. A confusion matirix [30] is used to define the performance of a classification system in a tabular format. In confusion matrix, the predicted values are on the horizontal axis, and the true values are on the vertical axis. For each class, it contains for values: true positive (TP), true negative (TN), false positive (FP), and false negative (FN).

For each class C, TP is defined as the number of truly detected objects in C. FP is the number of C objects that flasley clssified in another class. TN is the number of objects in other classes that are **not** classified in C. FN is the number of objects in other classes that are classified in C. Precision and recall are defined as follow using these values:

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

For each class C, precision measures the correctness of the model in identifying C objects. It is defined as the ratio between truly detected C objects to all objects detected as C. Whereas, recall is the ratio between the number of all detected C objects to all C objects. We use IoU (Intersection over Union) [31] criteria to calculate TP, FP, TN, and FN. The detected object is acceptable if the overlap between predicted and ground-truth bounding boxes is greater than IoU. IoU is defined as follow:

$$Precision = \frac{intersection\ area}{union\ area} \tag{3}$$

The Average Precision (AP) score is calculated by computing the area under the precision-recall curve. A larger area under the curve corresponds to a higher AP value and indicates better detection performance. Now, mAP metric is defined as follow:

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{4}$$

where AP_i is the AP for class i.

B. Model settings

The settings used in the models are given in the table I. *Number of Workers* is the number of parallel workers used during training. *Batch Size* is the number of training examples processed in a single iteration. Ismail et al. and FastV2 use the SGD optimizer, and YOLOv8 uses an auto selection between SGD and Adam based on the size of dataset and batch size. *Momentum* is a hyperparameter that accelerates SGD in the relevant direction. *Learning Rate* controls the amount of the changes of model hyperparameters in response to calculated error at each iteration. In YOLOv8, the default values of learning rate are different for SGD (0.01) and Adam (0.001) optimizers. *Detection Threshold* is the minimum confidence score required for a detected object to be considered as valid object detection. This threshold is set to 0.7 for all three models. All models use an image size of 640 pixels. IoU threshold is set to 0.7 for all three models. These settings help configure the training process for each model, influencing factors such as parallelism, optimization, and accuracy of object detection.

C. Results Analysis

All models were trained on an NVIDIA-SMI Tesla T4 with 16 GB of RAM. In the training phase, the best weights for the model was evaluated using mean Average Precision (mAP) at IoU thresholds equals to 0.5 (i.e. mAP@0.5). The model was validated based on the mAP at IoU thresholds ranging

TABLE I
MODEL SETTINGS

Model	Num worker	Batch size	Optim-izer	Mome-ntum	Learning rate	Detection Threshold	Image size	IoU
Ismail et al.	2	6	SGD	0.9	0.001	0.7	640	0.7
Fastv2	2	6	SGD	0.9	0.001	0.7	640	0.7
YOLOv8	8	16	Auto	0.937	Auto	0.7	640	0.7

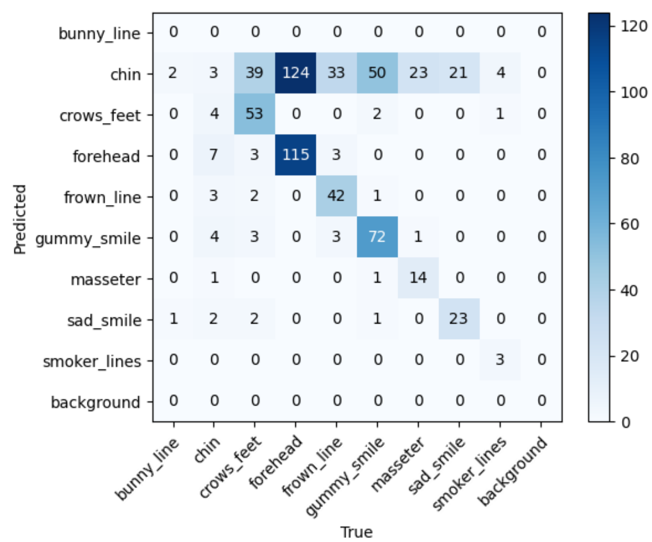

Fig. 8. Confusion matrix for esmail et al.

from 0.5 to 0.95 (mAP@0.5:0.95). Finally, the method was evaluated on the test set using mAP@0.5.

We employ the confusion matrix as an evaluation metric. The results of Ismail et al.'s model are depicted in Fig.8. Notably, this model demonstrates the highest level of accuracy in detecting smile line wrinkles, specifically the gummy smile. However, when compared to the Fastv2 and Yolov8 methods, it exhibits relatively lower overall accuracy. Nevertheless, it outperforms Fastv2 in identifying the forehead and sad smile categories. It is worth highlighting that a significant number of false detections in Fig.8 are attributed to chin wrinkles, which bear resemblance to other wrinkle types.

Fig.9 illustrates the confusion matrix outcomes for the Fastv2 model. According to this plot, the Fastv2 method surpasses Ismail et al.'s model in detecting bunny lines, chin, crow's feet, gummy smile, and masseter wrinkles, although it still falls short of the Yolov8 method. Fig.10 presents the confusion matrix for Yolov8 method. The outcomes indicate that the Yolov8 model exhibits markedly superior performance across all categories compared to the other models. Particularly, the Yolov8 model achieves the highest accuracy in detecting sad smile wrinkles, while showing relatively lower accuracy in identifying nose wrinkles. Moreover, it demonstrates a commendable level of accuracy in the remaining wrinkle categories.

In Fig. 11, we have graphically compared the performance of all models in detecting wrinkles using 6 sample images. In the first sample image, Ismail et al.'s method correctly identified two wrinkles, while the Fastv2 method failed to detect them. YOLOv8 also accurately detected forehead wrinkles with a larger detection area. In the second image, Ismail et al.'s method detected only two wrinkles, while Fastv2 and YOLOv8 detected 5 and 7 wrinkles, respectively. In the third image, Ismail et al.'s a method and Fastv2 detected nine wrinkles, while YOLOv8 accurately detected all 10 wrinkles. In the fourth to sixth images, YOLOv8 outperformed other methods by detecting more wrinkles. Thus, our approach benefits from higher recall and precision. These result are in conformance with Table II. Among the three methods, YOLOv8 achieved the highest precision, recall, and mAP@0.5 with only 11.2M parameters. On the other hand, the Ismail et al method had the highest number of parameters (41.3M) but had lower precision, recall, and mAP@0.5 values compared to YOLOv8. Fastv2 had a slightly lower parameter count (43.2M) than Ismail et al. Fastv2 results are lower than other methods.

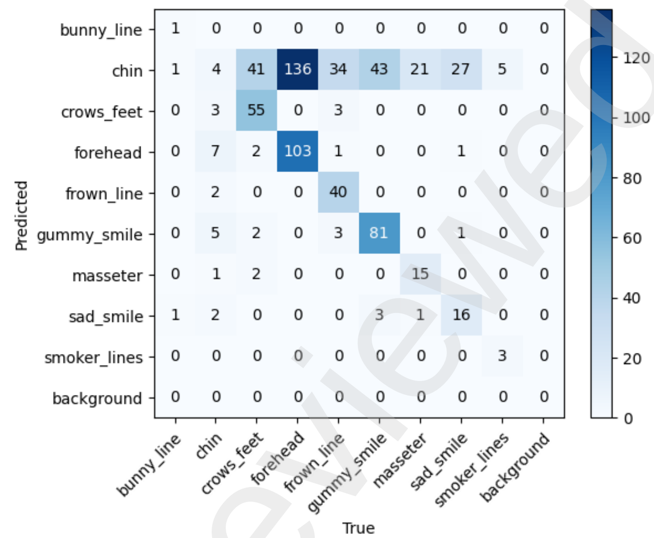

Fig. 9. Confusion matrix for Fastv2.

According to the results presented in Table III, YOLOv8 model exhibited impressive precision scores, ranging from 0.88 to 1, across various wrinkle classes. This implies that the model's predictions regarding the presence of wrinkles were highly accurate. The recall scores for all wrinkle classes were also notably high, ranging from 0.86 to 0.98, indicating that the model effectively detected wrinkles in the provided images. Furthermore, the model achieved commendable mAP@0.5 scores across different wrinkle classes, ranging from 0.8 to 0.98. These scores suggest that the model struck a good balance between precision and recall in the context of wrinkle detection. It demonstrated the model ability to effectively identify wrinkles while maintaining a satisfactory level of precision.

However, when considering stricter IoU thresholds (mAP@0.5:0.95), the model's performance showed a slight decline. The mAP@0.5:0.95 scores for all wrinkle classes ranged from 0.57 to 0.78, indicating a decrease in accuracy compared to the broader IoU range. Despite this reduction, the overall performance of the model remained reasonably accurate for wrinkle-detection tasks. In summary, the

TABLE II
COMPARISON OF PROPOSED METHOD

Method	Param	Images	Instances	Precision	Recall	mAP@0.5
Ismail et al.	41.3M	103	666	0.7	0.61	0.67
Fastv2	43.2M	103	666	0.65	0.56	0.62
YOLOv8	11.2M	103	666	0.95	0.86	0.92

TABLE III
RESULTS OF THE YOLOV8 EVALUATION ON THE TEST SET FOR EACH TYPE OF WRINKLES

Class	Images	Instances	Precision	Recall	mAP@0.5	mAP@(0.5:0.95)
all	103	666	0.95	0.86	0.92	0.69
bunny lines	103	666	1	0.6	0.8	0.5
chin	103	666	0.92	0.87	0.92	0.68
crows feet	103	666	0.97	0.97	0.98	0.74
forehead	103	666	0.88	0.87	0.92	0.7
frown line	103	666	0.92	0.9	0.95	0.73
gummy smile	103	666	0.97	0.98	0.99	0.78
masseter	103	666	0.89	0.97	0.96	0.77
sad smile	103	666	0.97	0.97	0.98	0.74
smoker lines	103	666	1	0.58	0.82	0.57

YOLOv8 model yielded high precision and recall scores across different wrinkle classes, achieving a good balance between precision and recall. Although there was a slight decrease in performance under stricter IoU thresholds, the model still exhibited a reasonable level of accuracy for wrinkle detection. These findings highlight the potential effectiveness of the YOLOv8 model in the field of wrinkle analysis and detection.

In addition, we presented a comparison of the validation curves for the implemented methods in Fig. 12, which demonstrates that YOLOv8 has better convergence compared to the other methods. Overall, these results demonstrate that the proposed YOLOv8 method is superior to other state-of-the-art wrinkle detection algorithms.

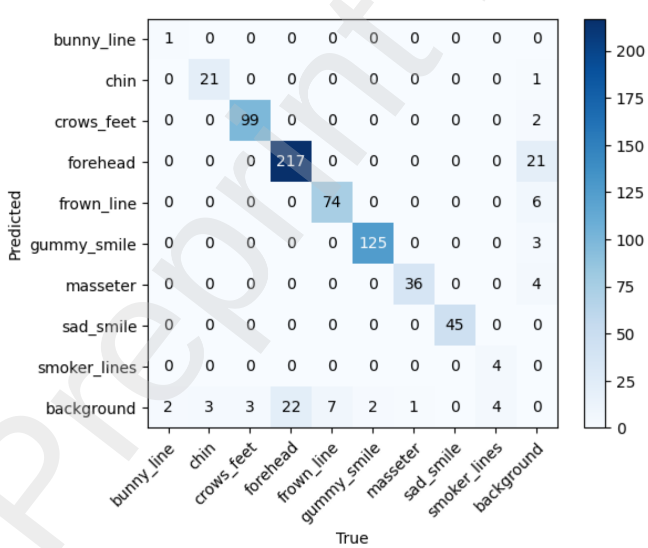

Fig. 10. Confusion matrix for Yolov8.

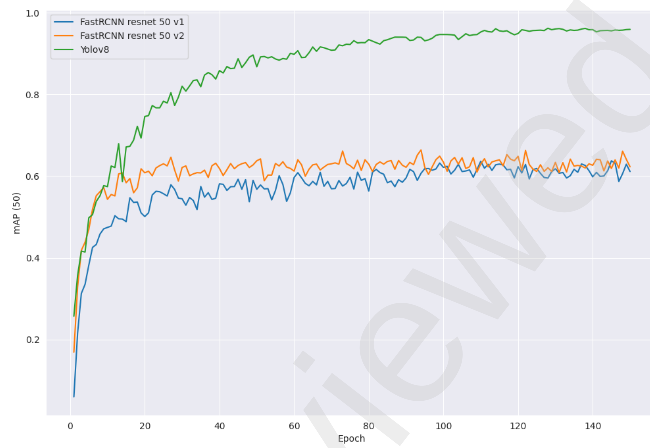

Fig. 12. Comparison of the validation curves.

VI. CONCLUSION

The deep learning-based automated wrinkle detection methods have attracted increasing attention in recent years due to their potential applications in fields such as skincare, cosmetics, and fashion. In this paper, we used Fast R-CNN ResNet-50v2 and YOLOv8 for wrinkle detection in facial images. We made a new dataset consisting of 1022 images of human faces with varying degrees of wrinkles. The annotations were made by an expert. The facial images were classified into nine different categories based on the wrinkle location on the face. By applying data augmentation techniques to our training data, the quantity of training data increased from 714 to 2138 images. The findings of this study have important implications for the development of automated wrinkle-detection methods, which can aid in the development of personalized skincare treatments. Future work can include the use of larger and more diverse datasets to validate the effectiveness of the proposed method.

REFERENCES

[1] M. H. Yap, N. Batool, C.-C. Ng, M. Rogers, and K. Walker, "A survey on facial wrinkles detection and inpainting: datasets, methods, and challenges," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 5, no. 4, pp. 505–519, 2021.

[2] G. Jocher, A. Chaurasia, and J. Qiu, "Yolo by ultralytics. 2023," *URL: https://github. com/ultralytics/ultralytics*, 2023.

[3] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[5] M. I. M. Ismail and A. N. Sung, "Acne lesion and wrinkle detection using faster r-cnn with resnet-50," in *AIP Conference Proceedings*, vol. 2676, no. 1. AIP Publishing LLC, 2022, p. 020007.

[6] T.-R. Chang and M.-Y. Tsai, "Classifying conditions of speckle and wrinkle on the human face: A deep learning approach," *Electronics*, vol. 11, no. 21, p. 3623, 2022.

[7] C.-C. Ng, M. H. Yap, N. Costen, and B. Li, "Wrinkle detection using hessian line tracking," *IEEE Access*, vol. 3, pp. 1079–1088, 2015.

[8] B. Çanak and M. E. Kamaşak, "Automatic scoring of wrinkles on the forehead," in *2017 25th Signal Processing and Communications Applications Conference (SIU)*. IEEE, 2017, pp. 1–4.

[9] S. Zaghbani, N. Boujneh, and M. S. Bouhlel, "Facial emotion recognition for adaptive interfaces using wrinkles and wavelet network," in *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*. IEEE, 2017, pp. 342–349.

Fig. 11. A visual example of the obtained results.

[10] W. Xie, L. Shen, and J. Jiang, "A novel transient wrinkle detection algorithm and its application for expression synthesis," *IEEE Transactions on Multimedia*, vol. 19, no. 2, pp. 279–292, 2016.

[11] R. M. Elbashir and M. Hoon Yap, "Evaluation of automatic facial wrinkle detection algorithms," *Journal of Imaging*, vol. 6, no. 4, p. 17, 2020.

[12] U. Sabina and T. K. Whangbo, "Edge-based effective active appearance model for real-time wrinkle detection," *Skin Research and Technology*, vol. 27, no. 3, pp. 444–452, 2021.

[13] E. Y. Tarasova and I. S. Mintii, "Web application for facial wrinkle recognition," in *CEUR Workshop Proceedings*, 2022, pp. 198–210.

[14] Q. Zheng, A. Purwar, H. Zhao, G. L. Lim, L. Li, D. Behera, Q. Wang, M. Tan, R. Cai, J. Werner *et al.*, "Automatic facial skin feature detection for everyone," *arXiv preprint arXiv:2203.16056*, 2022.

[15] G. Carlos da Silva, M. B. Barbosa, F. B. C. Júnior, P. L. Moreira, R. Werka, and A. A. Martin, "Detection of skin wrinkles and quantification of roughness using a novel image processing technique from a dermatoscope device," *Skin Research and Technology*, vol. 29, no. 6, p. e13335, 2023.

[16] S. Kim, H. Yoon, J. Lee, and S. Yoo, "Semi-automatic labeling and training strategy for deep learning-based facial wrinkle detection," in *2022 IEEE 35th International Symposium on Computer-Based Medical Systems (CBMS)*. IEEE, 2022, pp. 383–388.

[17] H. Deepa, S. Gowrishankar, and A. Veena, "A deep learning-based detection of wrinkles on skin," in *Computational Vision and Bio-Inspired Computing: Proceedings of ICCVBIC 2021*. Springer, 2022, pp. 25–37.

[18] A. K. Sameera, V. Samuktha, T. Akash, M. Sabeshnav, and S. Veni, "Real time detection of the various sign of ageing using deep learning," in *2022 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET)*. IEEE, 2022, pp. 38–43.

[19] H. Musafer and E. Tokgöz, *A Facial Wrinkle Detection by Using Deep Learning with an Efficient Optimizer*. Cham: Springer Nature Switzerland, 2023, pp. 381–388.

[20] G. Panis and A. Lanitis, "An overview of research activities in facial age estimation using the fg-net aging database," in *Computer Vision-ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part II 13*. Springer, 2015, pp. 737–750.

[21] O. F. Osman, R. M. I. Elbashir, I. E. Abbass, C. Kendrick, M. Goyal, and M. H. Yap, "Automated assessment of facial wrinkling: a case study on the effect of smoking," in *2017 IEEE international conference on systems, man, and cybernetics (SMC)*. IEEE, 2017, pp. 1081–1086.

[22] A. Savran, B. Sankur, and M. T. Bilge, "Comparative evaluation of 3d vs. 2d modality for automatic detection of facial action units," *Pattern recognition*, vol. 45, no. 2, pp. 767–782, 2012.

[23] F. Yanwei, "Fgnet,dataset," 2012.

[24] K. Ricanek and T. Tesafaye, "Morph: A longitudinal image database of normal adult age-progression," in *7th international conference on automatic face and gesture recognition (FGR06)*. IEEE, 2006, pp. 341–345.

[25] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, "The feret evaluation methodology for face-recognition algorithms," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 10, pp. 1090–1104, 2000.

[26] M. Minear and D. C. Park, "A lifespan database of adult facial stimuli," *Behavior research methods, instruments, & computers*, vol. 36, pp. 630–633, 2004.

[27] D. Tzutalin, "https://github.com/tzutalin/labelimg," 2015. [Online]. Available: https://github.com/tzutalin/labelImg

[28] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big data*, vol. 3, no. 1, pp. 1–40, 2016.

[29] M. Maity, S. Banerjee, and S. S. Chaudhuri, "Faster r-cnn and yolo based vehicle detection: A survey," in *2021 5th international conference on computing methodologies and communication (ICCMC)*. IEEE, 2021, pp. 1442–1447.

[30] B. P. Salmon, W. Kleynhans, C. P. Schwegmann, and J. C. Olivier, "Proper comparison among methods using a confusion matrix," in *2015 IEEE International geoscience and remote sensing symposium (IGARSS)*. IEEE, 2015, pp. 3057–3060.

[31] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 658–666.