Group-G: Age Classification Using Convolutional Neural Networks

A. Abstract

Automated human age classification (HAC) models utilize algorithms to estimate an individual's age based on their facial image. Despite its potential applications in various sectors like marketing and healthcare, age classification remains a complex problem for AI systems due to the challenges of facial image interpretation. Existing literature has employed deep CNN architectures like VGG-16, custom CNN models, and VGG-Face to classify age accurately, but these models are limited by high computational complexity and a need for significant training data. This report proposes a systematic approach using ResNet18, MobileNetV2, and ShuffleNetV2 models trained on APPA-REAL, UTKFace, and Adience datasets to address these challenges. The proposed methodology involves preprocessing the data, optimizing hyperparameters, and addressing class imbalance issues. The optimized models achieved an overall accuracy for age classification, with Resnet18 outperforming the other models on all datasets, demonstrating the effectiveness of the proposed methodology.

B. Introduction

Automated human age classification (HAC) models aim to estimate a person's age using information extracted from their facial image through dedicated algorithms. This technology finds application in various sectors, such as marketing and healthcare [1]. Age classification faces the challenge of shared similarities with other facial image interpretation tasks, which include face detection, feature extraction, and classification. The use of images for human age estimation or age group classification has extensive potential applications in age-invariant face identification, and face recognition across various age groups, as well as in commercial and law enforcement settings, security control and surveillance, e-learning, biometrics, human-machine interaction, and electronic customer relationship management. Age classification from human faces is a highly complex and challenging problem for artificial intelligence (AI) systems. This is because the appearance of a face can vary significantly due to various factors such as changes in pose, expressions, illumination, occlusions, image degradations caused by blur and noise, and other potential variables like makeup. In addition to these challenges, there are the additional complexities that come with aging, which is an incredibly intricate process that is extremely difficult to model accurately. Even when a group of people shares the same age, they may still look vastly different depending on various factors like their environment, lifestyle, and genes. These complexities make it incredibly challenging for AI systems to accurately classify an individual's age based on their facial features. Additionally, the scarcity of pre-classified datasets and the prevalence of low-quality images further complicate the task of age classification. However, despite the difficulties, age classification is a critical application of facial recognition technology in various industries like security, marketing, and healthcare.

To address the challenges of age classification in human faces, various methods and techniques have been developed. These include using advanced machine learning algorithms to effectively learn and model complex facial features. Additionally, large datasets of pre-classified images have been created to train these algorithms and improve their accuracy. Techniques such as data augmentation and normalization have also been employed to address the issues of low-quality images and variations in lighting and facial expressions. Furthermore, some methods use multiple features, such as facial texture and shape, to improve the accuracy of age classification [3]. Despite these efforts, the challenge of accurately classifying an individual's age from their facial features remains a complex problem in AI research. Existing literature suggests that deep CNN architectures like VGG-16, custom CNN models, and the VGG-Face model have been employed to tackle the challenges associated with age classification. While these approaches have demonstrated promising results, they also exhibit limitations, including high computational complexity and the requirement for substantial training data [2].

This report aims to address these challenges by proposing a systematic approach to optimize CNN models for age classification on the APPA-REAL[4, UTKFace[6][7], and Adience[5] datasets. To address these challenges, this report proposes a systematic approach using Convolutional Neural Networks (CNN) to classify facial images into distinct age groups. The methodology includes preprocessing datasets, manually removing low-quality images, optimizing hyperparameters, and employing techniques to prevent overfitting and address class imbalance issues. Preprocessing and cleaning of the images are performed prior to feeding them as input to train the CNN models using PyTorch, a deep-learning framework. The methodology involves using three datasets, which are partitioned into Train, Test, and Validation sets. The models used are ResNet18, MobileNetV2, and ShuffleNetV2, which are popular and efficient models for image classification. The CNN model is trained using the Adam optimization algorithm, a commonly used gradient-based optimization algorithm in deep learning. The implementation involved pre-processing the data by applying an 80-10-10 train/validation/test split and using image augmentation techniques to increase the size of the dataset and prevent overfitting. The CNN models were trained using the Adam optimization algorithm with a fixed batch size of 16 and the Cross-Entropy loss function. The optimized models achieved an overall accuracy for age classification, and the test accuracies for each dataset were reported. These results demonstrate the efficacy of the proposed methodology for training and optimizing CNN models for multi-class age classification.

B.1. Related Work

Initially, age estimation methods relied on calculating ratios between various facial feature measurements. Facial features such as eyes, nose, mouth, chin, wrinkles, etc., are located, and their sizes and distances are measured to determine the ratios, which are then used to categorize the face into different age groups based on handcrafted rules.[XXX] Hayashi [12] focused on the analysis of wrinkles and skin texture, employing the Hough transform as a method for studying these features. Zhou [13] utilized boosting for regression, while Geng [14] proposed the Aging Pattern Subspace method. More recently, Guo introduced an approach that extracts aging-related features through low-dimensional aging manifold obtained from subspace learning, enabling more accurate age estimation. In recent years, deep learning and CNN have shown great potential in feature learning and face recognition. They possess the capability to learn distinctive characteristic descriptors directly from image pixels [15], which are crucial for accurate age estimation. Widely recognized architectures, such as AlexNet, GoogLeNet, VGGNet, ResNet, SqueezeNet, and Xception, are often considered the most common due to their exceptional performance across various benchmarks, including age estimation tasks. Selim et al. [16] employed the k-nearest neighbor classifier and Local Binary Pattern for age classification. They trained their system using the FERET dataset, dividing the training data into five age groups and achieving an accuracy of 81%. Karthiyahani and Ridhar [17] utilized the canny edge detection method and elastic matching pattern for face recognition. They implemented a CNN with the backpropagation method on the ORL dataset, achieving 95% accuracy. Levi and Hassner [18] estimated age and gender using a deep neural network and a limited dataset. They tested their approach on the Adience benchmark, achieving an accuracy of 86.8%.

The field of age estimation using AI has seen significant advancements in recent years. Early methods relied on calculating ratios between different facial features to classify faces into age categories. More recent approaches use deep learning models to estimate age directly from facial images. However, accurate facial feature localization remains a challenging problem, making these methods unsuitable for in-the-wild images found on social platforms.

C. Methodology

C.1. Datasets

Three publicly available Face Age Classification pretrained image datasets from kaggle were used: UTKFace[6][7], Appa-Real[4] and Adience[5]. Originally UTKFace dataset is collected from various online sources, such as Google Image Search etc., Appa-Real is based on Apparent-real age rating and Adience is collected from flickr. Each dataset has its unique characteristics in terms of size, number of classes, age groups, and image resolution as shown in the table below. The UTKFace dataset is another widely used dataset for age classification. It consists of over 20,000 facial images of different individuals, with age ranging from 1 to 100 years old. The images were collected from publicly available sources, and each image is labeled with the subject's age, gender, and ethnicity. APPA-REAL Dataset is a collection of facial images of individuals from different age groups, races, and gender. The images were obtained from the APPA (Automated Pseudo-Photorealistic Portrait Generator) system, which was designed to generate facial images of different age groups based on statistical analysis of facial features. The Adience dataset is a collection of facial images for age and gender classification. The dataset includes the subject's age, gender, and the estimated age based on their appearance. The dataset is challenging due to the variations in the quality of images, lighting conditions, and poses.

Dataset Specifications	UTKFace	APPA-REAL	Adience
Total Number of Images	23,706	7,591	11,030
No of Classes	2	5	8
Image Resolution	200x200	816x816	-
Image Format	JPEG	JPEG	JPEG

Table 1. Specifications of all three datasets

Each dataset has multiple classes, and the standard deviation is calculated per class. The UTKFace dataset has two classes, and the standard deviation values are relatively low, with a mean of 0.223. The Adience dataset has eight classes, and the standard deviation values range from 0.213 to 0.243, with a mean of 0.226. The Appa-Real dataset has five classes, and the standard deviation values range from 0.249 to 0.260, with a mean of 0.256. A higher standard deviation means that the values in a class are more spread out, indicating that the class may be more difficult to classify accurately.

All the Datasets were split in the ratio 80:10:10 for Training, Validation and Testing. Applied preprocessing and filtering steps such as resizing the images to a common size and normalizing the pixel values. Furthermore, Cross-validation has been performed on the three models for each

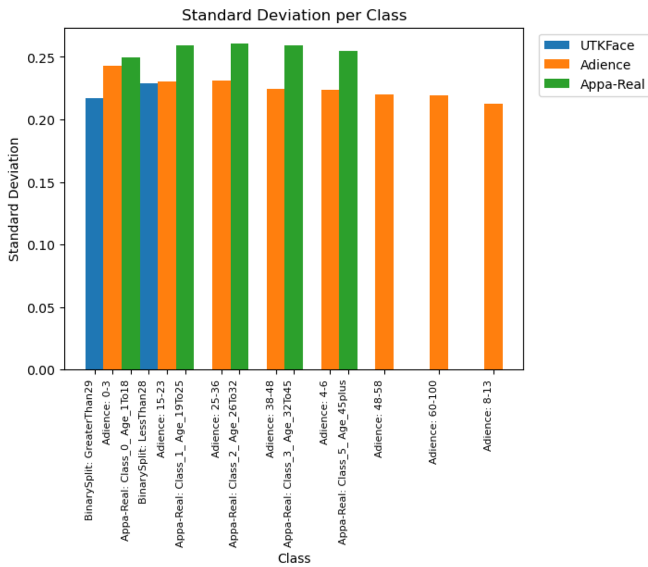

Figure 1. Standard Deviation per Class for each Dataset

dataset. Overall, the complexity of the class representations varied across the datasets, with Adience and Appa Real being more challenging due to the multi-class nature of the problems.

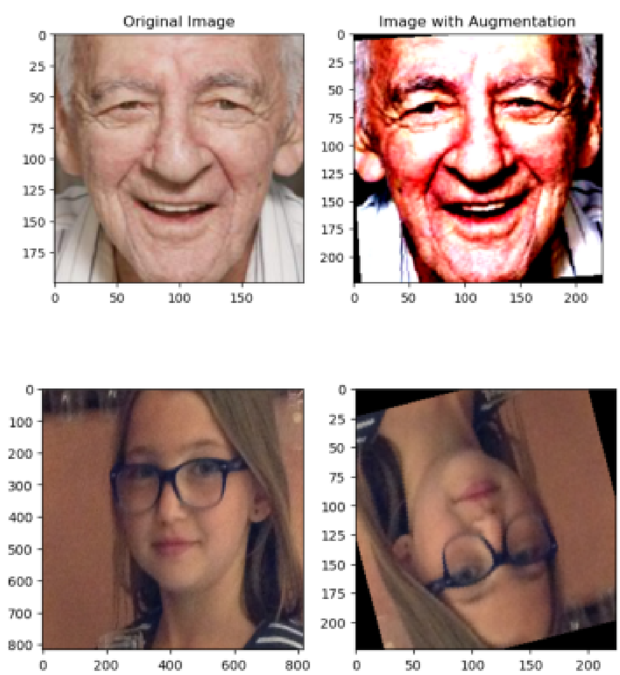

Figure 2. Sample Image From UTKFace and Adience Dataset

C.2. CNN Models

The selected models are ResNet18, MobileNetV2, and ShuffleNetV2. These models are well-known and widely used in image classification tasks due to their strong performance and relatively low computational complexity.

ResNet18 is a deep residual network that uses skip connections to improve the flow of information through the network. This architecture is suitable for image classification tasks where deeper networks tend to perform better. Additionally, ResNet18 has a relatively low computational complexity compared to other deeper architectures, making it suitable for training on lower-end hardware [8]. MobileNetV2 is a lightweight neural network architecture designed for mobile and embedded vision applications. It uses depthwise separable convolutions, which significantly reduce the number of parameters and FLOPS required for training, making it suitable for deployment on mobile and edge devices with limited computational resources [9]. ShuffleNetV2 is another lightweight neural network architecture designed to reduce the computational complexity of traditional convolutional neural networks. It uses group convolutions and channel shuffling to reduce the number of FLOPS required for training while maintaining high accuracy [10]. Overall, the selected models are suitable as they offer a balance between computational complexity and accuracy. ResNet18 is suitable for deeper image classification tasks while MobileNetV2 and ShuffleNetV2 are suitable for lightweight and mobile applications. This allows us to choose the most appropriate model for our specific use case while still achieving high accuracy. The computational complexities of the selected CNN models for training and validation phases in terms of wall clock time for one-epoch training and number of FLOPS calculations are as follows:

For training and validation, the ResNet18, MobileNetV2, and ShuffleNetV2 models have been compared using the UTKFace, Adience, and Appa-Real datasets. For one-epoch training, ResNet18 required around 990.436 billion FLOPS, MobileNetV2 required around 4018.565 billion FLOPS, and ShuffleNetV2 required around 393.202 billion FLOPS. For 50 epochs, ResNet18 took around 4393.40 seconds (87sec/epoch) for training and validation on the UTKFace dataset, 8469.67 seconds (169sec/epoch) on the Adience dataset, and 2074.58 seconds (41sec/epoch) on the Appa-Real dataset. MobileNetV2 took approximately 5397.47 (107sec/epoch) seconds for training and validation on the UTKFace dataset, 8146.36 (162sec/epoch) seconds on the Adience dataset, and 2418.18 seconds (48sec/epoch) on the Appa-Real dataset. ShuffleNetV2 took approximately 4730.70 seconds (94sec/epoch) for training and validation on the UTKFace dataset, 7965.39 seconds (159sec/epoch) on the Adience dataset, and 2264.93 seconds (45sec/epoch) on the Appa-Real dataset.

Overall, ShuffleNetV2 requires the least number of FLOPS for training and validation, making it computationally efficient. ResNet18 and MobileNetV2 require higher FLOPS and longer training times, making them more computationally intensive. However, ResNet18 provides a higher level of accuracy than ShuffleNetV2 and is a more suitable choice for applications that require higher accuracy at the expense of computational efficiency. MobileNetV2 offers a good balance between accuracy and computational efficiency and is a suitable choice for applications that require a good level of accuracy without sacrificing too much in terms of computational complexity. The train set was used to train the model, while the validation set was used to tune the hyperparameters and prevent overfitting. The test set was used to evaluate the final performance of the model on unseen data. Additionally, cross-validation was also used during the hyperparameter tuning process to ensure the model's generalization performance.

C.3. Optimization Algorithm

In order to train our CNN model for the classification task, Utilized the Adam optimization algorithm, which is a commonly used gradient-based optimization algorithm in deep learning [11]. A learning rate of 0.01 has been employed, and a fixed batch size of 16 due to limited computational resources. The Cross-Entropy loss function, which is widely used for multi-class classification tasks. To optimize our CNN model, Used a range of hyperparameters, including the learning rate, batch size, and loss function. Utilized the Cross-Entropy loss function for multi-class classification and a fixed batch size of 16 due to limited computational resources. Adam as the optimization algorithm for training the model, which adapts the learning rate for each parameter and combines the advantages of AdaGrad and RMSProp.

To validate and optimize our model. The training set was used to train the model, the validation set was used to tune hyperparameters, and the test set was used to evaluate the final performance of the model on unseen data. During training, Monitored the model's performance on the validation set and adjusted hyperparameters accordingly. Followed an iterative process of training and validation, starting with a baseline model using default hyperparameters and then systematically varying the hyperparameters to find the optimal values. A manual approach was used to tune the learning rate. In this approach, the learning rate was gradually increased from a 0.001 value to a 0.1 while monitoring the training loss. The learning rate was then set to the value at which the training loss started to increase again, indicating that the model had reached its optimal point. Finally, Evaluated the performance of our optimized CNN models using several metrics, including precision, recall, F1 score, and confusion matrix.

D. Results

The experiment setup began with pre-processing the input data, where an 80-10-10 train/validation/test split was applied. To increase the size of the dataset and prevent overfitting, image augmentation techniques such as random rotations, flips, and shifts were performed. For training the CNN model, the Adam optimization algorithm was used with a fixed batch size of 16 and the Cross-Entropy loss function. A random search was performed over a range of learning rates, and it was observed that a learning rate of 0.01 produced satisfactory results for all three datasets.

The performance of the optimized CNN models was evaluated using several metrics such as ROC/AUC, confusion matrix, precision, recall, and F1 score, which provided a comprehensive understanding of the model's classification accuracy and misclassification of different classes. Our optimized CNN models achieved an overall accuracy for age classification, and the test accuracies for each dataset were as follows: Resnet18 achieved an accuracy of 83.31% for UTKFace dataset, 53.59% for Adience dataset, and 42.86% for Appa-Real dataset. MobileNetV2 achieved an accuracy of 79.07% for UTKFace dataset, 46.07% for Adience dataset, and 23.68% for Appa-Real dataset. ShuffleNetV2 achieved an accuracy of 81.59% for UTKFace dataset, 52.06% for Adience dataset, and 38.91% for Appa-Real dataset. These results demonstrate the efficacy of the proposed methodology for training and optimizing CNN models for multi-class age classification. The proposed methodology, which involved pre-processing the data using image augmentation techniques and training the models using the Adam optimization algorithm.

D.1. Main Results

Trained and optimized three CNN models: Resnet18, MobileNetV2, and ShuffleNetV2 for multi-class age classification on three different datasets: UTKFace, Adience, and Appa-Real.

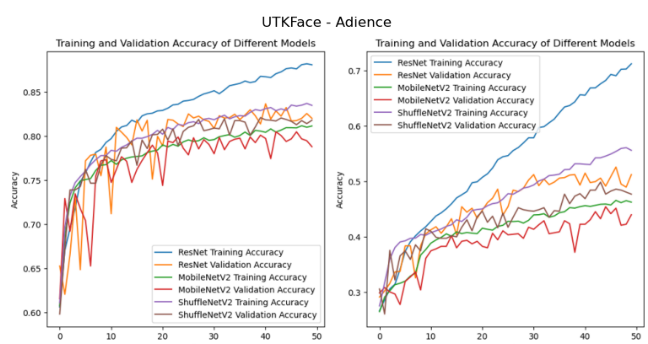

Figure 3. Train and Validation Accuracy Plot

All three graphs show the training and validation accuracy of different CNN models as the number of epochs increases during the training phase. The UTKFace dataset's graph shows that the ResNet18 model has the highest accuracy for both training and validation sets. The accuracy for both sets increases with each epoch and plateaus after around 15-20 epochs. The validation accuracy is consistently lower than the training accuracy, but the gap between them reduces as the number of epochs increases. In the Adience dataset's graph, the ResNet18 model has the highest accuracy for the training set, but the MobileNetV2 model has the highest accuracy for the validation set. The accuracy of both sets increases with each epoch, but the validation accuracy plateaus earlier than the training accuracy, at around 10 epochs. Again, the validation accuracy is consistently lower than the training accuracy, but the gap between them reduces as the number of epochs increases. The Appa-Real dataset's graph shows that the ResNet18 model has the highest accuracy for both training and validation sets. The accuracy for both sets increases with each epoch and plateaus after around 15-20 epochs. The validation accuracy is consistently lower than the training accuracy, but the gap between them reduces as the number of epochs increases.

In the learning rate tuning process, we started with a coarse search using a logarithmic scale of learning rates ranging from 0.001 to 0.1. Based on the initial results, we performed a finer search with learning rates ranging from 0.001 to 0.1. We found that a learning rate of 0.01 gave the best performance across all three datasets and three CNN models.

Model	Resnet18	MobileNetV2	ShuffleNetV2
UTKFace	83.31%	79.07%	81.59%
Adience	53.59%	46.07%	52.06%
Appa-Real	42.86%	23.68%	38.91%

Table 2. Test Accuracy Comparison of Resnet18, MobileNetV2, and ShuffleNetV2 on Different Datasets

From Table 2. The Resnet18 model generally outperformed the other models on all datasets, achieving the highest test accuracy on UTKFace, Adience, and Appa-Real datasets. MobileNetV2 performed better than ShuffleNetV2 on all datasets, but both models had lower test accuracies compared to Resnet18.

This difference in performance can be attributed to the deeper architecture of Resnet18 compared to MobileNetV2 and ShuffleNetV2, which allows it to learn more complex features and patterns from the data. Additionally, Resnet18 has been shown to have better performance on a range of computer vision tasks compared to MobileNetV2 and ShuffleNetV2. Overall, the results demonstrate the importance of selecting an appropriate model architecture for the given task and dataset, as well as the effectiveness of the proposed methodology for optimizing CNN models for age classification. Based on the results obtained that, the UTKFace dataset seems to be the best for age classification. The Resnet18 model achieved the highest accuracy on this dataset with 83.31%, which is significantly better than the other two models. On the Adience dataset, all three models achieved much lower accuracy, with Resnet18 achieving the highest at 53.59%. This could be due to the larger age range of this dataset and the greater variability in image quality. On the Appa-Real dataset, the Resnet18 model also achieved the highest accuracy with 42.86%. However, the overall performance on this dataset was lower compared to UTKFace, which could be due to the greater variability in image quality. Based on the ROC curves, we can ob-

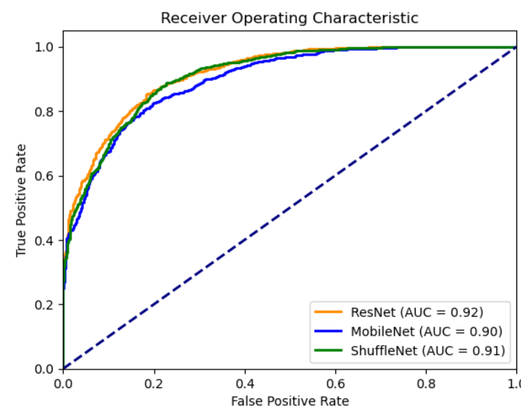

Figure 4. ROC Curve of UTKFace on 3 Models

serve that Resnet18 outperformed MobileNetV2 and ShuffleNetV2 on all three datasets. Resnet18 had the highest AUC scores for UTKFace, Adience, and Appa-Real, indicating better overall performance in terms of the tradeoff between true positive rate and false positive rate. MobileNetV2 had the lowest AUC scores across all datasets, indicating that it had the poorest performance among the three models. ShuffleNetV2 had slightly better performance than MobileNetV2 on Adience and Appa-Real, but still performed worse than Resnet18. Overall, the ROC curves suggest that Resnet18 is the best performing model among the three models for age classification on these datasets. It appears that the UTKFace ResNet and MobileNetV2 models performed similarly in terms of accuracy, with both achieving an accuracy score of around 0.83. The ShuffleNetV2 model had a slightly lower accuracy score of 0.81, but still performed reasonably well. In contrast, the Adience and Appa-Real datasets had lower accuracy scores across all three models. The Adience dataset had an accuracy score of 0.53 for the ResNet model and 0.46 for the MobileNetV2 model, while the ShuffleNetV2 model achieved an accuracy score of 0.51. The Appa-Real dataset had accuracy scores

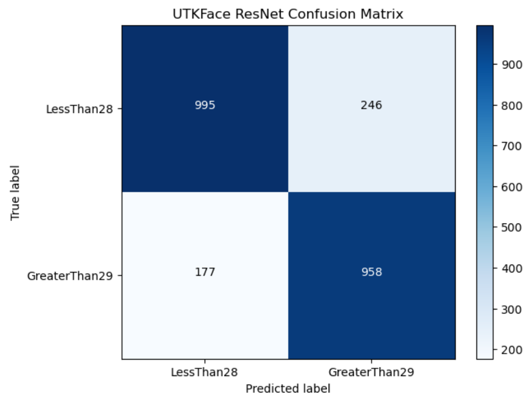

Figure 5. Confusion Matrix of UTKFace

ranging from 0.29 to 0.42 across all three models. Looking at the confusion matrices for the UTKFace dataset, we can see that all three models perform well in correctly identifying the age range of individuals. However, there are some differences in performance between the models. Looking

Model	Precision	Recall	F1-Score	Accuracy
UTKFace ResNet	0.83	0.83	0.83	0.83
UTKFace MobileNetV2	0.81	0.79	0.79	0.79
UTKFace ShuffleNetV2	0.82	0.81	0.81	0.81
Adience ResNet	0.53	0.53	0.52	0.53
Adience MobileNet	0.45	0.46	0.38	0.46
Adience Shufflenet	0.50	0.51	0.50	0.51
Appa-Real ResNet	0.41	0.43	0.41	0.42
Appa-Real MobileNetV2	0.11	0.23	0.13	0.23
Appa-Real ShuffleNetV2	0.38	0.39	0.37	0.38

Table 3. Classification report of all Datasets (Weighted Avg.)

at the UTKFace dataset, we can see that all three models (ResNet, MobileNetV2, and ShuffleNetV2) have performed well with an F1-score of around 0.81-0.83. Among these three models, MobileNetV2 has the lowest precision score, but it has a higher recall than ResNet and ShuffleNetV2. For the Adience dataset, all three models (ResNet, MobileNet, and ShuffleNet) have similar precision scores, but their recall and F1-score are relatively low. For the Appa-Real dataset, ResNet has the highest precision score, but the lowest recall and F1-score among the three models. In contrast, MobileNetV2 has the highest recall score but the lowest precision score. ShuffleNetV2 has a similar F1-score as ResNet but has a slightly higher precision and recall score. In summary, ResNet and ShuffleNetV2 have performed well on the UTKFace and Appa-Real datasets, while MobileNetV2 has a higher recall score on the UTKFace dataset. However, all three models have relatively low recall and F1-score on the Adience dataset. Therefore, the performance of these models depends on the dataset they are applied to.

Comparision of classification reports for UTKFace dataset with and without transfer learning. The results show that transfer learning has generally improved the performance of the models. Specifically, for ResNet, both precision and recall have improved for the ”LessThan28” class with transfer learning, while precision and recall for the ”GreaterThan29” class remained almost the same. For MobileNetV2, precision and recall for the ”LessThan28” class have decreased, but the model showed a significant improvement in predicting the ”GreaterThan29” class. For ShuffleNetV2, precision and recall for the ”LessThan28” class have slightly increased, while precision and recall for the ”GreaterThan29” class have significantly improved with transfer learning. Overall, transfer learning has improved the performance of the models for the UTKFace dataset, especially in predicting the ”GreaterThan29” class, while the ”LessThan28” class shows mixed results.

For Appa-Real Dataset, Transfer learning improved the test accuracy of ResNet, MobileNetV2, and ShuffleNetV2 models for AppaReal dataset. The ResNet model had the highest accuracy at 39.79%, followed by ShuffleNetV2 at 41.52%, and MobileNetV2 at 26.08%. Without transfer learning, the models had lower accuracy rates.

D.2. Ablative Study

Early stopping is a regularization technique that prevents overfitting in machine learning models during training. By monitoring the model's performance on a validation set, early stopping can halt the training process before the model starts overfitting on the training data. As a result, the model converges faster, and the training time is significantly reduced (on average, by 20%). This is achieved by avoiding unnecessary iterations that do not contribute to the model's improvement, which enables the model to converge faster. Therefore, early stopping helps the model converge faster and reduces training time significantly by halting the training process early and avoiding unnecessary iterations.

References

1. Fu, Y., Guo, G., Huang, T. S. (2010). Age synthesis and estimation via faces: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(11), 1955-1976.
2. Yin, J., Zhang, Y., Hu, X., Chen, X. (2017). Multitask deep neural network for age estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 10-18).
3. Lanitis, A., Draganova, C., Christodoulou, C. (2004). Comparing different classifiers for automatic age estimation. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 34(1), 621-628.
4. Agustsson, E., Timofte, R., Van Gool, L. (2017). Apparent and real age estimation in still images with deep residual regressors on APPA-REAL database. In Proceedings of the IEEE International Conference on Automatic Face Gesture Recognition (pp. 1-8).
5. Eidinger, E., Enbar, R., Hassner, T. (2014). Age and gender estimation of unfiltered faces. IEEE Transactions on Information Forensics and Security, 9(12), 2170-2179.
6. Y. Wang, T. Wu, A. Coates, and A. Y. Ng, "End-to-End Facial Landmark Detection with Stacked Hourglass Networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5689-5698.
7. R. Zhang, P. Isola, and A. A. Efros, "Colorful Image Colorization," in European Conference on Computer Vision, 2016, pp. 649-666.
8. He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 770-778).
9. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.-C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4510-4520).
10. Ma, N., Zhang, X., Zheng, H.-T., Sun, J. (2018). ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design. In Proceedings of the European Conference on Computer Vision (pp. 116-131).
11. Kingma, D. P., Ba, J. (2014). Adam: A Method for Stochastic Optimization. arXiv preprint arXiv:1412.6980.
12. Hayashi, J. I., Yasumoto, M., Ito, H., Koshimizu, H. (2002). Age and Gender Estimation Based on Wrinkle Texture and Color of Facial Images. International Conference on Pattern Recognition, 2002. Proceedings (Vol.1, pp.405-408 vol.1). IEEE
13. Zhou, S. K., Georgescu, B., Zhou, X., Comaniciu, D. (2010). Method for performing image based regression using boosting. US, US7804999
14. Geng, X., Zhou, Z. H., Zhang, Y., Li, G., Dai, H. (2006). Learning from facial aging patterns for automatic age estimation. ACM International Conference on Multimedia, Santa Barbara, Ca, Usa, October (pp.307-316). DBLP
15. Liu, W., Wen, Y., Yu, Z., Yang, M. (2017a). Large-Margin Softmax Loss for Convolutional Neural Networks. Proceedings of the 34th International Conference on Machine Learning, 70, 507-516
16. Selim, M., Raheja, S., Stricker, D. (2015). Real-time human age estimation based on Facial images using uniform Local Binary Patterns. In VISAPP (2), International Conference on Computer Vision Theory and Applications, 2015 Proceedings (Vol. 2, pp. 1-8).
17. Karthigayani, P., Sridhar, S. (2013). Detection tree-based occlusion detection in face recognition and estimation of human age using back propagation neural network. Journal of Computer Science and Technology, 10(1), 115–127
18. Levi, G., Hassner, T. (2015). Age and gender classification using convolutional neural network. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 34-42.

E. Supplementary Material

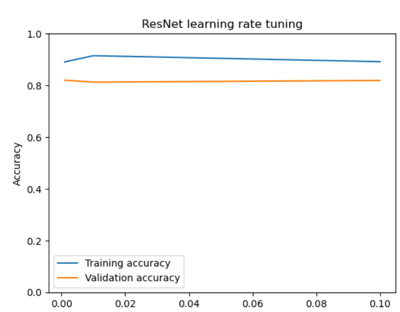

Figure 6. LR Tuning on RESNET UTKFace

[this section is appended to the main report draft]: You may include appendices to your final report to support different sections of the main draft.

Contents