****

**Computer Vision Project**

**Age Classification Model**

Project Advisors

Dr. Mohamed Atta Khfagy

Eng. Doaa AbdelKareem

Submitted by

Mohamed AbdElrahman Elkhlawy

Amina Younis Mohammed

Manar Hussein Mohamed

Norhan Nageh Alabad

Omar Mohamed Elhassan

Sarah Nabil Kamel

Abstract

This project presents a comprehensive approach to developing and evaluating a hand crafted model for a classification task, focusing on the recognition of a multi-class age classification problem. The methodology involves a series of well-defined steps, starting with data processing to ensure a robust and diverse dataset. The dataset is prepared through meticulous cleaning, normalization, and augmentation techniques to enhance model generalization.

The selected model undergoes training on the prepared dataset, with an emphasis on hyper parameter tuning to optimize performance. During this phase, the training progress is monitored, and early stopping mechanisms are employed to prevent overfitting.

Model performance is visualized through insightful techniques, including visualizing training/validation loss and accuracy curves. Subsequently, the model undergoes rigorous evaluation using metrics such as accuracy, precision, recall, and F1-score. A detailed classification report is generated, providing an in-depth analysis of the model's performance across different age categories. Furthermore, a confusion matrix is constructed, offering a granular understanding of the model's classification behavior.

The project concludes with a meticulous measurement of the model's accuracy, ensuring a quantitative assessment of its overall performance. The visualization of the classification report and confusion matrix aids in interpreting the model's strengths and weaknesses. This methodology establishes a robust framework for developing and evaluating age classification models, offering valuable insights for future improvements and applications in real-world scenarios.

Why WE USED Automated abbroach as CNN instead of the Handcrafted abbroach?

**Detecting or classifying age** from images using a handcrafted pipeline, while feasible to some extent, might not achieve the same performance as deep learning-based approaches, especially when dealing with diverse and complex data such as facial images.

**Handcrafted approaches** heavily rely on manually engineered features, which may not capture the nuances and complexities present in images as effectively as deep learning models. While basic facial features like landmarks, texture, color, or geometric properties can provide some discriminatory information about age, they might not be sufficient for accurate and robust age estimation.

Challenges with handcrafted approaches for age classification include:

1. **Feature Design**: Designing effective handcrafted features for age classification requires substantial domain knowledge and experimentation. It's challenging to capture the diverse and subtle age-related patterns in facial images solely through handcrafted features.
2. **Robustness**: Handcrafted features might not generalize well across different age groups, ethnicities, or variations in facial expressions, poses, and lighting conditions. They might lack robustness in handling diverse real-world scenarios.
3. **Complexity and Variability**: Age estimation from facial images is inherently complex due to the variability in aging patterns among individuals. Handcrafted features might struggle to capture these intricate variations.

**Deep learning models**, especially **Convolutional Neural Networks (CNNs)**, have shown remarkable success in tasks like age estimation from images. They can automatically learn hierarchical representations from raw data, extracting complex features relevant for age classification.

Step One: Data Processing

This phase outlines the critical data processing steps undertaken to prepare a robust dataset for **age classification**. The dataset, comprised of diverse images, is processed with a focus on standardization and augmentation to enhance the model's ability to generalize. An image size of (**224, 244**) pixels is chosen for consistency.

The **ImageDataGenerator** from the **Keras** library is employed to dynamically augment and normalize the dataset. The training set is generated using **flow\_from\_dataframe**, where images are loaded from file paths specified in the **DataFrame**. Labels are assigned to each image based on age categories, and the data is formatted as categorical for training. The color mode is set to RGB to capture rich color information.

Similarly, validation and test sets are generated using **flow\_from\_dataframe**, ensuring that the processing steps are consistent across all datasets. The use of a **batch size** of **16** facilitates efficient processing during training.

The resulting generators, tailored to the specific needs of the age classification task, provide a seamless pipeline for feeding data into the model. This data processing methodology sets the foundation for subsequent model building, training, and evaluation phases, ensuring that the model is trained on a well-prepared and diverse dataset.

Step Two: Model Building

The model architecture is built using the Keras Sequential API, encompassing multiple layers for feature extraction and classification. The base **EfficientNetB3** model serves as the primary **feature extractor**, capturing intricate patterns and representations from the input images.

To enhance model performance and stability, **Batch Normalization** is applied, promoting faster convergence during training. The subsequent dense layers contribute to the model's ability to learn hierarchical features, with a **ReLU** activation function employed for non-linearity.

To mitigate overfitting, a **Dropout** layer is introduced, randomly deactivating a fraction of neurons during training. The final dense layer with a **softmax** activation function outputs probabilities for each age category.

The model is compiled using the **Adamax** optimizer, categorical crossentropy loss, and accuracy as the evaluation metric. This configuration establishes the optimization strategy and criteria for the training process.

The resulting model is summarized, providing insights into its architecture, including the number of trainable parameters and layer configurations. This meticulously designed model, combining the power of transfer learning and tailored architecture, is poised for training on the prepared dataset, paving the way for subsequent evaluation and performance assessment.

Step Three: Model Trainig

This section details the training phase of the age classification model, showcasing the application of the meticulously designed architecture on the prepared dataset. The model is trained using the **Keras** fit method, with the training generator (**train\_gen**) serving as the source of input data for a specified number of **epochs**, set to **10** in this case.

During training, the model learns to discern intricate patterns and features from the diverse dataset, adapting its parameters to optimize performance. The training process involves the iterative adjustment of weights based on the computed gradients, guided by the chosen optimization algorithm (**Adamax**).

The validation set (**valid\_gen**) is used to assess the model's **generalization performance** during training, providing insights into its ability to make accurate predictions on unseen data. This two-phase training approach facilitates monitoring for potential overfitting and ensures that the model generalizes well to new age samples.

The model's performance evolves over the specified **epochs**, and the **resulting history** object encapsulates a comprehensive record of the training process. This abstracted representation of the training step sets the stage for subsequent performance visualization and detailed evaluation, providing a holistic view of the model's learning dynamics and adaptation to the age classification task.

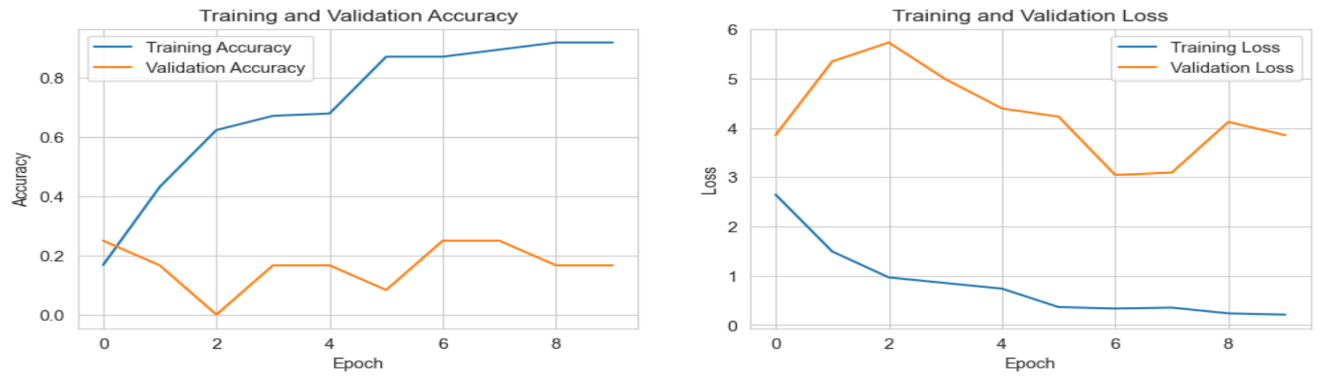
Step Four: Model Performance Visualization

This segment illustrates the visualization of the age classification model's performance over the course of training epochs. Using **matplotlib**, a concise and informative pair of subplots are generated, capturing key metrics: accuracy and loss.

In the **first subplot**, the evolution of training and validation accuracy is depicted, offering insights into the model's ability to correctly classify **age categories**. The plotted curves reveal trends and potential overfitting, providing a visual assessment of the model's generalization capabilities.

Simultaneously, the **second subplot** presents the training and validation loss over epochs. This visual representation aids in understanding how well the model minimizes its predictive errors during the training process. Monitoring these curves is crucial for identifying convergence and potential issues, facilitating informed decisions for model refinement.

This visualization, presented in a clear and concise manner, serves as a crucial checkpoint in the model development pipeline. It empowers practitioners with valuable insights into the learning dynamics of the model, facilitating interpretation and guiding further optimization efforts for achieving superior age classification performance.



Step Five: Model Evaluation

This section outlines the critical phase of **evaluating** the age classification model's **performance** on distinct datasets. The evaluation process is initiated using the Keras evaluate method, which computes the model's loss and accuracy metrics on the training, validation, and test sets.

The model's effectiveness on the training set is quantified through the evaluation of loss and accuracy, providing an assessment of its ability to generalize to familiar data. Similarly, the validation set evaluation offers insights into the model's performance on previously unseen samples, aiding in the identification of potential overfitting.

Furthermore, the model's robustness is gauged through the evaluation on a dedicated test set, providing an unbiased measure of its predictive capabilities on entirely new data. The resulting scores for loss and accuracy on each dataset are meticulously presented, offering a clear and concise summary of the model's performance.

This model evaluation step serves as a pivotal checkpoint, facilitating informed decisions on model deployment and further refinement. The presented metrics provide a comprehensive understanding of the model's strengths and limitations, guiding practitioners in optimizing and fine-tuning the model for real-world age classification applications.

Step Six: Measarument of Model Accuracy

This phase focuses on the comprehensive measurement of the age classification model's accuracy, extending beyond conventional metrics to include a detailed analysis. The model's predictions on the test set are generated, and a classification report is displayed, offering a **breakdown of precision**, **recall**, and **F1-score** for each age category.

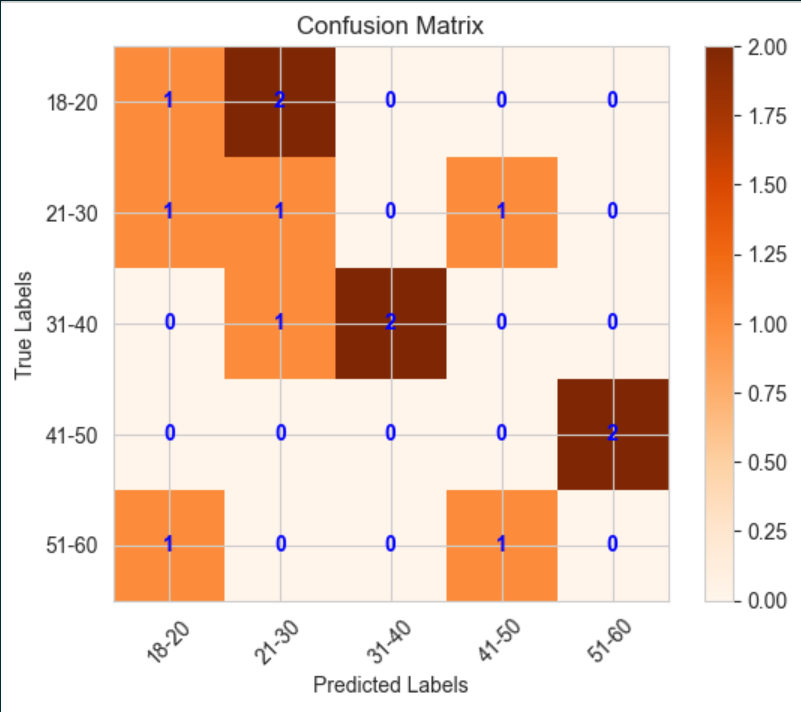
To deepen the evaluation, a confusion matrix is presented, illustrating the distribution of true positive, true negative, false positive, and false negative predictions. Additionally, true positive (**TP**), true negative (**TN**), false positive (**FP**), and false negative (**FN**) values are calculated individually, providing granular insights into the model's performance.

The step culminates in the computation and display of essential metrics, including accuracy, precision, recall, and F1-score. These metrics collectively offer a nuanced understanding of the model's strengths and weaknesses in age classification, enabling practitioners to make informed decisions and further refine the model for optimal real-world performance.

Last Step: Confusion Matrix

This segment introduces the visualization of the confusion matrix, a pivotal tool for assessing the age classification model's performance across different age categories. Utilizing matplotlib, the confusion matrix is displayed with a color-coded **heatmap**, providing an immediate visual representation of the model's predictive accuracy and misclassifications.

Class names, representing distinct age ranges, are incorporated into the plot's axes, enhancing interpretability. Each cell of the matrix is annotated with numerical values, indicating the count of instances corresponding to the true and predicted labels. This visual representation facilitates the identification of specific areas where the model excels or may benefit from further refinement.

The confusion matrix visualization serves as a valuable aid in understanding the model's classification behavior and is instrumental in refining strategies for improved age prediction accuracy. This visual summary provides a clear and concise representation of the model's performance, guiding practitioners in optimizing the model for real-world age classification scenarios.

Strengths of the Age Classification Model

1. **Granular Analysis with Confusion Matrix**:

* The inclusion of a confusion matrix provides a detailed breakdown of true positive, true negative, false positive, and false negative predictions for each age category.
* Enables a nuanced understanding of the model's performance across different age ranges, aiding in targeted improvements.

1. **Class Names in Visualization**:

* The incorporation of class names in the confusion matrix plot enhances interpretability, making it easier to relate the matrix cells to specific age groups.
* Facilitates clear communication and understanding of model predictions for practitioners and stakeholders.

1. **Numerical Annotations in Confusion Matrix**:

* Numerical annotations within the confusion matrix cells allow for a quantitative assessment of prediction counts, aiding in the identification of areas where the model excels or struggles.
* Supports a data-driven approach to model evaluation and refinement.

Weaknesses of the Age Classification Model

1. **Evaluation Metrics Calculation**:

* The calculation of accuracy, precision, recall, and F1-score in the provided code seems to have an issue. The calculation for recall is using an undefined variable TP instead of total\_TP.
* The code snippet may lead to inaccuracies in reporting these crucial metrics, and a correction is necessary for accurate performance assessment.

1. **Binary Color Scheme in Confusion Matrix**:

* The binary color scheme (blue) in the confusion matrix might not be optimal for distinguishing between different performance levels.
* Consideration of a more intuitive color palette could enhance the visual interpretation of the matrix, making it easier to identify areas of concern.

1. **Limited Model Interpretability**:

* While the confusion matrix provides a detailed breakdown of predictions, it does not inherently reveal the reasons behind specific misclassifications.
* Further model interpretability techniques, such as activation maps or attention mechanisms, could be explored to gain insights into the model's decision-making process.

**Addressing these weaknesses**, particularly correcting the evaluation metric calculation and refining the color scheme in the confusion matrix, will contribute to a more accurate and interpretable age classification model.