Age Estimation from Facial Photos: A CNN-Based Approach with Multi-Model Feature Fusion

Zixuan Lyu

Beijing Aidi International School Beijing, China

* Corresponding Author Email: aLzx3081@outlook.com

Abstract. This study was designed to improve the accuracy and efficiency of facial age recognition by training and tuning advanced deep learning models. This study evaluated the performance using the UTK Face dataset and multiple deep learning models for the facial age recognition task and found that a three-model fusion (EfficientNetB0, DenseNet121, and Inception V3) had the best accuracy and generalization ability. The study also tried adding a Transformer layer, but the results were not significantly improved. Triple Model Fusion (EfficientNetB0, DenseNet121, and Inception V3) with the addition of the Transformer layer had an overall accuracy of 64%, which was slightly lower than the version without the Transformer (65% accuracy). Triple Model Fusion performed in all tested configurations Best with 65% accuracy.

Keywords: age estimation, image recognition, CNN, feature fusion, transformer.

1. Introduction

Age is one of the most critical pieces of information about an individual. Facial age recognition is a task that poses great difficulty for human beings. In many fields, the use of facial age recognition to accomplish specific tasks is of paramount importance. For instance, in criminal cases, the accurate age of an unknown offender's face is needed to obtain the individual's identity information; in forensic medicine, the victim's age must be determined quickly to gather information about the victim. Therefore, using models to achieve facial age recognition can lead to more precise age predictions.

In this task, the selection of models and algorithms is crucial in ensuring the accuracy of the predictions. The choice of models and algorithms depends on the performance of different models in facial recognition tasks. More accurate age recognition can bring unprecedented possibilities in areas like public safety, medical diagnosis, and personalized services. For example, models can enable the police to acquire the identity information of criminals more easily, or swiftly obtain information about the victim in forensic medicine [1]. However, the potential risk is that those possessing this technology may more easily gain others' age information. Hence, the responsible use of this technology is vitally important.

Facial age recognition is not only an extremely challenging task but also has broad applicability across many fields. This task is often treated as a classification problem, requiring the use of advanced deep learning architectures, particularly Convolutional Neural Networks (CNNs). In existing deep learning research, there are many outstanding models that can be applied to image recognition, including facial age recognition. The EfficientNet model is a method of automatically adjusting the size of various parts of the neural network to achieve higher accuracy with less computation [2]. Its adjustment method is by continually expanding the network's depth, width, and resolution. The ResNet model mitigates the vanishing gradient problem through the introduction of "skip connections" [3]. This innovative structure enables the network to be trained deeper, thus capturing more complex features. DenseNet 's every layer is connected to all previous layers, promoting feature reuse [4]. This close connection improves gradient flow and enhances feature propagation. Inception captures information at different scales through the "Inception module" [5]. It can capture features at multiple scales, enhancing the network's expressiveness. VGG network is known for its simple and uniform structure, using a repetitive structure with small convolution kernels [6]. Despite its simplicity, it performs excellently in image recognition tasks. These models each have advantages and characteristics in facial age recognition, collectively providing robust support for age recognition.

For example, EfficientNet 's automatic adjustment ability makes it highly accurate in resource-limited environments, especially suitable for facial age recognition in mobile devices or embedded systems; ResNet's skip connections allow training of deeper structures to capture complex age features, while DenseNet's tight connections foster fine analysis of minute features; Inception's multi-scale capturing ability can identify different scale features that faces may exhibit at various ages; VGG's small convolution kernels capture local features, aiding in the detailed analysis in facial age recognition.

The primary objective of this study is to enhance the accuracy and efficiency of facial age recognition through training and tuning advanced deep learning models. By exploring and comparing the performance of different models in facial age recognition tasks, this study hope to find a more optimal solution. The successful implementation of this technology will not only assist in the fields of public safety for identity recognition and criminal investigation but also play a significant role in medical diagnosis, personalized services, and other aspects.

2. Method

2.1. Characterization and Preprocessing of the UTK Dataset

This study employs the UTK Face dataset [7], a large-scale and diverse collection of facial images. The dataset is versatile, suitable for a variety of applications including age, gender, and race recognition. This study utilizes a pre-cropped subset of the dataset that also includes age labels, encompassing a total of 23,708 images. Each image in this subset is a color image utilizing the RGB color mode.

1) Dataset preprocessing procedure: All preprocessing steps described in this section have been meticulously designed to function under limited computational resources. The primary objective is to ensure the dataset's quality and consistency, thus laying a solid foundation for the subsequent training and evaluation of models.

2) Image reading and pixel normalization: The OpenCV library is employed for reading images from the UTK Face dataset. Given that OpenCV operates under a default BGR color space, a conversion to RGB was implemented to align with the preferences of most deep learning models. Concurrently, pixel value normalization was conducted, reducing the input pixel range to [0,1], thereby facilitating rapid and robust model convergence.

3) Age label discretization and classification: This study designs an age label classification scheme, inspired by "Age Detection Model using CNN: A Complete Guide" [8]. The scheme transforms continuous age values into a series of predefined discrete categories, simplifying the model's output space while aligning more closely with real-world application needs. Figure 1 shows a preview of the training set after being labeled.

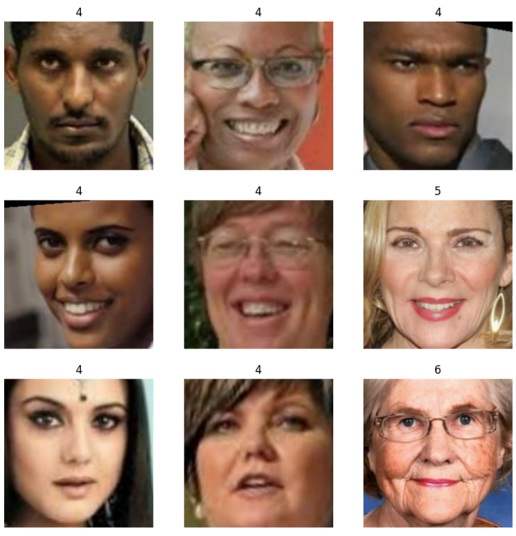

Fig. 1 UTK Face training set after classifying age labels.

4) Optimization of dataset structure: Utilizing Python's built-in shutil library, the pre-processed images are sorted by age categories and stored in corresponding subdirectories. This step significantly streamlines the subsequent batch processing of data using TensorFlow and enhances the code's maintainability.

5) Randomized division of training and validation sets: This study leverages the "image_dataset_from_directory" function from the TensorFlow library [9] to randomly partition the image dataset into training and validation sets at an 80-20 ratio. This ratio is based on empirical evidence, capable of achieving comprehensive model training under limited computational resources while retaining sufficient data for model validation.

6) Confirmation and Visualization of Data Morphology: To visually confirm the dataset characteristics and label accuracy, the matplotlib library is employed for data visualization. Additionally, it is verified that the shape of the data aligns with the model's input specifications.

2.2. Training of Efficient-Net, ResNet, DenseNet, Inception, VGG on UTK Dataset

The aim of this section is to train five different deep learning models, EfficientNetB0, ResNet50, DenseNet121, Inception and VGG16, on the UTK dataset. The purpose of doing so is to assess these five models on the UTK dataset and to try to select the better performing model among them for more optimization.

2.3. Mathematical Formulas and Evaluation Metrics

This study uses several evaluation metrics to quantify and compare the performance of different models on the UTK dataset. This section provides detailed mathematical definitions and explanations of these evaluation metrics.

1) Accuracy: Accuracy refers to the proportion of instances that the model has correctly predicted out of the overall sample count. Mathematically, the Accuracy can be expressed as:

$$\text{Accuracy} = \frac{|\text{TP}|+|\text{TN}|}{|\text{TP}|+|\text{TN}|+|\text{FP}|+|\text{FN}|} \tag{1}$$

2) Precision: Precision refers to the proportion of instances that the model accurately forecasts under the positive classification and are indeed in the positive category to the number of samples that the model estimates to belong to the positive category. Mathematically, the Precision can be expressed as:

$$\text{Precision} = \frac{|\text{TP}|}{|\text{TP}|+|\text{FP}|} \tag{2}$$

3) Recall: Recall is the proportion of the total instances identified by the model as falling within the positive category and indeed in the positive category to the number of instances that are in the positive category. Mathematically, the Recall Rate can be expressed as:

$$\text{Recall} = \frac{|\text{TP}|}{|\text{TP}|+|\text{FN}|} \tag{3}$$

4) F1-Score: The F1-Score is the reconciled mean of precision and recall and employed to consider both precision and recall. Mathematically, the F1-score can be expressed as:

$$\text{F1} - \text{score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Presion} + \text{Recall}} \tag{4}$$

5) Sparse categorical cross entropy loss: The use of sparse categorical cross entropy loss as a loss function in multiclassification problems has certain advantages, especially when the labels have some kind of continuity or sequentially.

In the specific context of this study, this study is dealing with the UTK dataset, where the labels are characterized by some continuity or orderliness. This means that the labels are not just categorized independently of each other and in a non-ordered way, but also have some kind of inherent sequential relationship.

The sparse categorical cross-entropy loss is similar in mathematical form to the general categorical cross-entropy loss, but it allows us to directly input these labels with sequential relationships in integer form without the need for solo thermal coding. This reduces computational complexity and memory requirements, especially when the number of categories is large. The mathematical representation is as follows:

$$\text{SparseCategorical Cross} - \text{entropy Loss} = -\sum_{i=1}^{N} y_i \log(\hat{y}_i) \quad (5)$$

Where:

- y_i is the true label of sample (expressed as an integer).
- $\hat{y}_i$ is the probability that the model predicts that sample belongs to the real category y_i.
- N is the number of samples.

Since age labels have continuity, this loss function setting can better capture this intrinsic relationship between labels, which may help the model learn more accurate feature representations.

In summary, considering the continuity of labels and computational efficiency, this study chooses to use utilizing sparse categorical cross-entropy for the loss metric.

2.4. Model Performance Evaluation

1) Efficient-net performance evaluation: Table I. shows the performance of Efficient-Net model on UTK dataset. The EfficientNetB0 model performs well on the UTK dataset, with an overall accuracy of 60%. Especially on category 0, its precision and recall are as high as 83% and 90%, respectively. However, the performance of the model is relatively weak on category 2 and category 3.

Table 1. Efficient-Net Performance

	Precision	**Recall**	**F1-score**	**support**
0	0.83	0.90	0.87	293
1	0.68	0.62	0.65	277
2	0.47	0.59	0.52	380
3	0.56	0.58	0.57	1099
4	0.58	0.62	0.60	1532
5	0.64	0.46	0.54	798
6	0.71	0.68	0.69	362
Accuracy			0.60	4741
Macro avg	0.64	0.64	0.64	4741
Weighted avg	0.61	0.60	0.60	4741

2) Evaluation of ResNet performance: Table II shows the performance of Resnet model on UTK dataset. According to Table II, the overall accuracy of the ResNet50 model is 59%. On category 0, the precision was excellent at 94% but 51% on recall, respectively. However, the model performs relatively poorly when processing images in categories 1 and 3.

Table 2. ResNet Performance

	Precision	Recall	F1-score	Support
0	0.94	0.51	0.66	293
1	0.51	0.62	0.56	277
2	0.57	0.43	0.49	380
3	0.54	0.67	0.60	1099
4	0.58	0.63	0.60	1532
5	0.62	0.56	0.59	798
6	0.80	0.52	0.63	362
Accuracy			0.59	4741
Macro avg	0.65	0.56	0.59	4741
Weighted avg	0.61	0.59	0.59	4741

3) Evaluation of DenseNet performance: The overall accuracy of the DenseNet121 model was 62%. The model performed relatively well on category 0, 1 and category 6, but poorly on category 2.

Table 3 DenseNet Performance

	Precision	Recall	F1-score	Support
0	0.90	0.84	0.87	293
1	0.71	0.55	0.62	277
2	0.51	0.57	0.54	380
3	0.59	0.53	0.56	1099
4	0.59	0.70	0.64	1532
5	0.64	0.50	0.56	798
6	0.70	0.71	0.71	362
Accuracy			0.62	4741
Macro avg	0.66	0.63	0.64	4741
Weighted avg	0.62	0.62	0.62	4741

4) Inception performance assessment: Table 4. shows the performance of Inception model on UTK dataset.

Table 4. Inception Performance

	Precision	Recall	F1-score	Support
0	0.91	0.77	0.83	293
1	0.62	0.52	0.57	277
2	0.56	0.56	0.56	380
3	0.60	0.43	0.50	1099
4	0.56	0.71	0.63	1532
5	0.60	0.52	0.56	798
6	0.61	0.78	0.68	362
Accuracy			0.60	4741
Macro avg	0.64	0.61	0.62	4741
Weighted avg	0.61	0.60	0.59	4741

The overall accuracy of the Inception V3 model on the UTK dataset was 60%. The model performed well on category 0 and category 1, but poorly on category 2 to category 6.

5) VGG performance assessment: Table 5. shows the performance of VGG model on UTK dataset.

The VGG16 model performed relatively worst with an overall accuracy of only 32%. In particular, the precision and recall are almost zero on every category except category 4.

6) Summary: Figure 2. are the accuracy and parameters in histogram of five models.

Table 5. VGG Performance

	Precision	Recall	F1-score	Support
0	0.00	0.00	0.00	293
1	0.00	0.00	0.00	277
2	0.00	0.00	0.00	380
3	0.00	0.00	0.00	1099
4	0.32	1.00	0.49	1532
5	0.00	0.00	0.00	798
6	0.00	0.00	0.00	362
Accuracy			0.32	4741
Macro avg	0.05	0.14	0.07	4741
Weighted avg	0.10	0.32	0.16	4741

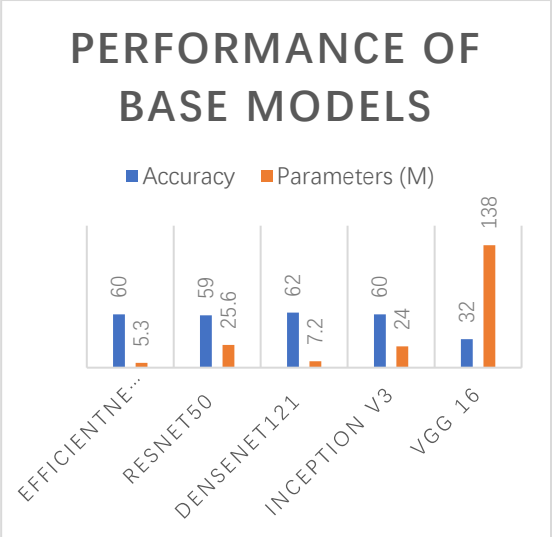

Fig. 2 Histogram of the performance of the base model

Combining the above evaluations, the EfficientNetB0, DenseNet121 and Inception models demonstrate high accuracy on the UTK dataset, so this study chooses these three models for the next step of model fusion. Although the performance of ResNet50 is also relatively impressive, its accuracy on categories 1 and 6 is slightly lower than the above three models and relatively higher parameters. In addition, this study decided not to use the VGG16 model because it not only has a low accuracy rate, but also has too many model parameters, resulting in high computational costs.

In the next phase, this study will focus on how to effectively integrate the three models, EfficientNetB0, DenseNet121 and Inception, in the expectation of further improving the accuracy and generalization ability of the models.

2.5. Three-model Fusion

1) Three-model fusion model structure: This study analyzes the performance of three base models, EfficientNetB0, DenseNet121, and Inception V3, based on the UTK dataset, and fuse the features of these three models to construct a new model (see Figure 3.). Each base model is first passed through its own pre-trained network for feature extraction, then passed through a

GlobalAveragePooling2D layer and a Dropout layer for downscaling and regularization, and finally passed through a fully connected layer to output 32-dimensional features.

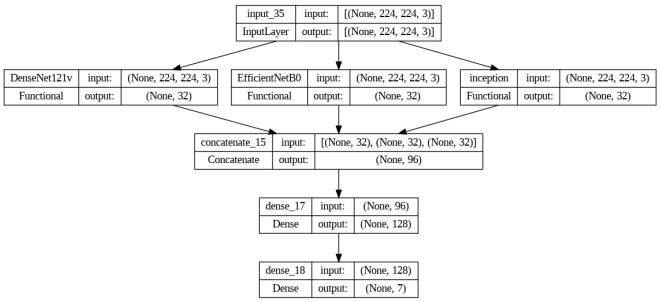

Fig. 3. Basic structure of Three-model fusion model

The feature outputs of these three models are concatenated together and the information is integrated through a fully connected layer with 128 dimensions. The resulting output from this fully connected layer is subsequently passed to an output layer for age classification. This structure allows the model to fully utilize the strengths of the three different models to achieve better recognition results.

2) Three model fusion + transformer structure: In Xiaofei Zhang's study [10] on attentional modeling for cataract diagnosis, the study was conducted by adding an attentional layer on top of three pre-integrated models to identify and focus on the abnormal regions inside the eye where cataract lesions occur more accurately. Inspired by this study, this study not only integrates multiple image features, but further introduces a Transformer layer after feature integration. (See Figure 4.). This study used the Transformer structure as originally introduced in [11]. In this structure, the feature outputs of the three previously fused models are stacked and passed into the Multi-Head Attention layer of the Transformer, which allows the model to acquire feature information more efficiently at different levels and aspects. The Transformer layer also includes a feed-forward neural network and Layer Normalization to further enhance the generalization ability of the model.

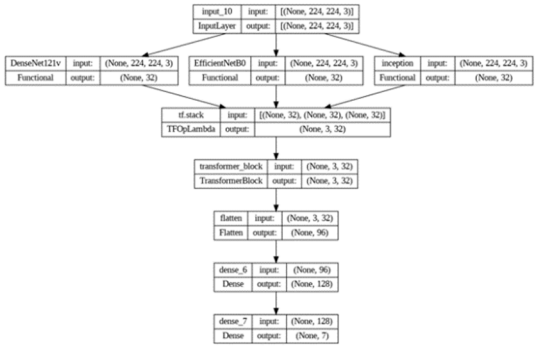

Fig. 4. Three model fusion + Transformer model structure

3) Hyper parameterization of the model: The hyperparameters used in this study include learning rates of 0.001, 0.0001 and 0.00001 tested in the base fusion model (the optimal learning rate was chosen for the next step of training in the subsequent model optimization), a loss function of "SparseCategoricalCrossentropy", and an optimizer of Adam, Dropout rates of 0.1, fully connected dimensions of 32 and number of attention heads 2. And Checkpoint was added to the callback during training so that the model weights that performed best on the validation set during training could be saved. Because of the added checkpoint, this study set the model training at 5epochs. These settings were chosen based on preliminary experiments and literature review.

4) Advantages of transformer: The Transformer structure has proven its efficiency and flexibility within the domain of natural language processing. In this task, Transformer can provide more complex feature interactions and powerful representation learning capabilities. Through the multi-head self-attention mechanism, the model can capture higher-order feature interactions between different models. Since the Transformer structure uses a fully connected layer internally, this allows

it to learn more abstract and complex representations. By introducing Transformer, the model proves its effectiveness by obtaining higher accuracy on the age recognition task.

3. Results

3.1. Evaluation of Three-Model Fusion with Varied Learning Rates

To deeply investigate the effect of different learning rates on the training effectiveness of the basic triple fusion model (see Figure 3.), three different learning rates were selected in this study: 0.001, 0.0001, and 0.00001. For each learning rate setting, this study performed training for five epochs and recorded the accuracy of the model under each epoch on both the training set and the validation set.

5) *Learning rate of 0.001:* Table 6. demonstrates the training and validation performance of the triple fusion model when the learning rate is set to 0.001.

Table 6 Basic structure of Three-model fusion model in learning rate 0.001

Epochs	Accuracy	Val accuracy
1	0.5570	0.6184
2	0.6391	0.6288
3	0.6811	0.6313
4	0.7215	0.6317
5	0.7597	0.6429

At this learning rate, the model performs relatively more robustly on both the training and validation sets, especially at high epoch values, and the validation accuracy maintains a relatively stable level. After 5 epochs of training, the validation accuracy of the model is 65%. The validation accuracy is relatively higher than the basic models. This means combining three models can have a better accuracy than single model. At the same time, this could also be since the complexity of the model becomes higher.

6) *Learning rate of 0.0001:* Table 7. demonstrates the training and validation performance of the triple fusion model when the learning rate is set to 0.0001.

The model in this setting shows a gradual increase in training accuracy. However, at the same time, there is an unstable change in validation accuracy, which is usually a sign of overfitting. After 5 epochs of training, the validation accuracy of the model is 62%.

Table 7 Basic structure of Three-model fusion model in learning rate 0.0001

Epochs	Accuracy	Val accuracy
1	0.5634	0.6136
2	0.7060	0.6210
3	0.8169	0.6220
4	0.8837	0.6205
5	0.9074	0.6186

7) *Learning rate of 0.00001:* Table 8. demonstrates the training and validation performance of the triple fusion model when the learning rate is set to 0.00001.

Table 8. Basic structure of Three-model fusion model in learning rate 0.00001

Epochs	Accuracy	Val accuracy
1	0.4739	0.5294
2	0.6214	0.5623
3	0.7114	0.5790
4	0.8040	0.5893
5	0.8777	0.5849

At this learning rate, although the model's accuracy during training gradually increases, the validation accuracy is lower and fluctuates little, which may be due to the learning rate being too small, resulting in the model converging too slowly. After 5 epochs of training, the validation accuracy of the model is 59%.

8) Model training results of Three-model fusion model

Table 9. Three-model fusion model Performance

	Precision	**Recall**	**F1-score**	**Support**
0	0.90	0.88	0.89	293
1	0.74	0.66	0.70	277
2	0.57	0.54	0.56	380
3	0.60	0.58	0.59	1099
4	0.62	0.69	0.65	1532
5	0.67	0.59	0.63	798
6	0.73	0.73	0.73	362
accuracy			0.65	4741
macro avg	0.69	0.67	0.68	4741
weighted avg	0.65	0.65	0.65	4741

a) Performance metrics: accuracy, loss function values, F1 scores: Table IX shows that the performance of the model varies on different categories. The overall accuracy reached 0.65, which is a moderate result. In terms of loss function, the loss value of the model in the first cycle (Epoch 1) is 1.0350, which is reduced to 0.5708 in the fifth cycle (Epoch 5), indicating that the model is gradually optimized during the training process. The average value of F1 score is 0.68, which is a further evidence of the model's more robust overall performance.

b) Performance analysis of each category: After analyzing the performance of each category in detail, this study find that there are obvious differences in the performance of the model on different categories. Specifically, category 0 performs the best, having the highest F1 score of 0.89, which indicates that the model has superior recognition ability on this category. This is followed by category 1 and category 6, which also show relatively good performance with F1 scores of 0.70 and 0.73, respectively. However, category 2, category 3, category 4 and category 5 all have F1 scores below 0.7, especially category 2 and category 3, and these low scores indicate that the model's performance on these categories needs to be improved.

c) Visualization of the training process: As shown in Figure. 6, the performance of the model on both the training and validation sets indicates significant performance improvement. For the training set, the accuracy of the model gradually improves from the initial 0.5570 to 0.7597, while the loss value is reduced from 1.0350 to 0.5708, both improvements demonstrating significant optimization of the model during the learning process. The situation is similar on the validation set, where the accuracy improves from 0.6184 to 0.6429, which is a smaller magnitude but still shows that the model has some generalization ability. However, it is worth noting that the loss value on the validation set increased in the fourth cycle, from 0.8622 to 0.9330, and then decreased to 0.9063 in the fifth cycle, which could be an early sign of overfitting. Therefore, it is also important to focus on strategies to prevent overfitting in future model optimization.

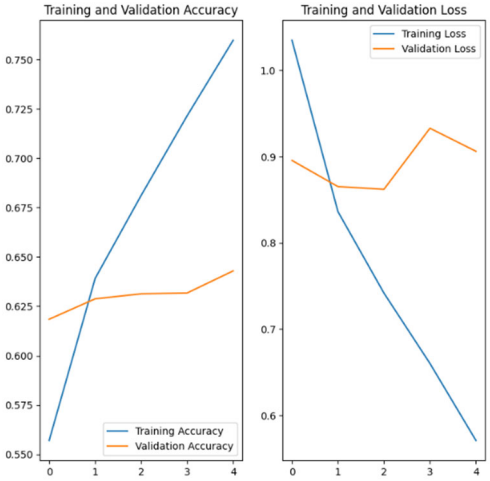

Fig. 5. Triple Fusion Model Visualization of the model training process

d) Computational complexity and latency: In the hardware configuration, each training cycle takes an average of 650 seconds, with each step taking about 1 second. The latency of the model predictions averaged 0.29 seconds, which is fast enough for most real-time applications.

9) Optimal learning rate selection: After comprehensively comparing the model performance under three different learning rates, this study decided to choose 0.001 as the preferred learning rate. Although there is a certain degree of overfitting risk for the model under this setting, its combined performance on the training and validation sets is still relatively good.

3.2. Training Results of Triple Fusion Model + Transformer

10) Overall performance metrics: A brief overview of the overall performance on the validation and test sets. (Including a portion of hyperparameters, learning rate 0.001)

11) Triple model fusion + transformer.

Table 10. Training Results of Triple Fusion Model Performance

	Precision	Recall	F1-score	Support
0	0.92	0.77	0.84	293
1	0.66	0.57	0.61	277
2	0.53	0.58	0.55	380
3	0.60	0.49	0.54	1099
4	0.59	0.77	0.67	1532
5	0.69	0.57	0.63	798
6	0.81	0.66	0.73	362
accuracy			0.64	4741
macro avg	0.69	0.63	0.65	4741
weighted avg	0.65	0.64	0.64	4741

a) Performance metrics: accuracy, loss function value, F1 score: The parameter set uses a three-model fusion with Transformer structure, and the learning rate is set to 0.001. In terms of the overall performance of the model, the accuracy is 0.64, the macro-averaged F1 score is 0.65, and the weighted-average F1 score is 0.64. This shows that the model has a relatively well-balanced performance on the classification task.

b) Performance analysis by category: Table X shows the performance of varied categories: in particular, category 0 and category 6 exhibited higher values of 0.84 and 0.73 on the F1 score, respectively. On the contrary, categories 2 and 3 have F1 scores of only 0.55 and 0.54, suggesting room for optimization. It is worth mentioning that category 4 achieves the highest recall of 0.77 despite its F1 score of 0.67. Compared to the non-Transformer structure, there is only a 0.01

difference in the precision, which may stem from random factors during the training process, etc., and further optimization of the hyperparameters is needed.

c) Visualization of the training process: As shown in Figure. 7, the training set accuracy improves from 0.5508 to 0.7376 while the validation set accuracy improves from 0.6009 to 0.6273 within 5 training cycles. However, the loss function value on the validation set shows an increase after a gradual decrease from 0.8679 to 0.9236, which may be an indication of overfitting.

d) Computational complexity and latency: The average latency of the model for a single prediction was approximately 119 ms. this latency can be roughly attributed to the addition of the Transformer layer, which correspondingly increased the computational complexity of the model.

Fig. 6. Visualization of the triple fusion model training process

3.3. Comparative Analysis of Model Performance

12) Tabulated comparison of model metrics.

13) Comprehensive performance analysis: The models' performance varies across different architectural designs, with the Triple Model Fusion showing marginal supremacy with respect to Accuracy as well as F1 Score measurements. However, upon integrating a Transformer layer into the architecture, the model's performance manifests a slight decline, especially in macro-averaged F1 score. The scores of models are summarized in Table XI.

14) Significance of accuracy and F1 score metrics: The elevated performance metrics of the Triple Model Fusion, concerning both accuracy and F1 score, establish it as a relatively robust model for classification tasks. The metrics serve as a litmus test for the model's ability to correctly identify multiple classes, while balancing precision and recall effectively.

Table 11. Model Evaluation Comparison

	EfficientNetB0	**DenseNet121**	**Inception V3**	**Triple Model Fusion**		**Triple Model Fusion + Transformer**
Accuracy	60%	62%		60%	65%	64%
F1 Score (macro average)	0.64	0.64		0.62	0.68	0.65
F1 Score (weighted average)	0.60	0.62		0.59	0.65	0.64

15) Complexity versus performance trade-off: The inclusion of the Transformer layer escalates the computational complexity while minutely affecting performance. This raises an imperative

question concerning the justifiability of this additional complexity vis-à-vis the achieved performance metrics.

16) *Fusion versus individual architectures:* It's discernible that a fusion of multiple architectures (Triple Model Fusion) provides a performance edge over singular architectures. However, this edge diminishes when a Transformer layer is added, indicating that the benefit of fusion is partly negated by the complexity of the Transformer layer.

3.4. Recommendations for Future Research

Category-wise Performance: For categories demonstrating lower performance metrics, data augmentation or solutions to class imbalance should be investigated.
Overfitting: Considering potential signs of overfitting, regularization methods like L1/L2 or Dropout layers should be explored.

Transformer Layer Tuning: Given that the research focus is on Triple Fusion and Transformer, more granular tuning of Transformer parameters may unearth untapped performance potential.

4. Discussion

4.1. Restatement of Conclusion and Achievements

The study undertook the development and evaluation of different architectures, including EfficientNetB0, DenseNet121, and Inception V3, as well as the fusion of multiple models (Triple Model Fusion) with and without a Transformer layer. Among these configurations, the Triple Model Fusion displayed the best performance with an accuracy of 65% and macro- and weighted-average F1 scores of 0.68 and 0.65, respectively. The inclusion of a Transformer layer did not substantially improve the performance, instead showing a slight decline in the macro-averaged F1 score.

4.2. Limitations of the Study

Time Constraint: The duration of the study did not allow for exhaustive hyperparameter tuning, which may have limited performance optimization.

Computational Resources: The available computational power limited the number of architectures and ensemble methods that could be evaluated.

Model Complexity and Hyperparameter Tuning: The study involved a wide range of models and configurations, which made it challenging to optimize each model comprehensively.

4.3. Potential Solutions for Limitations

Time Management: Future studies could be designed as a multi-phase project, allowing adequate time for hyperparameter tuning.

Resource Allocation: Employing cloud-based or distributed computing can make it feasible to explore a broader set of architectures and ensemble methods.

Focused Optimization: Subsequent research could focus on optimizing a select few promising architectures, such as the Triple Model Fusion with Transformer layers.

4.4. New Questions and Recommendations for Future Research

Effect of Augmentation Techniques: What role could data augmentation play in improving the performance of categories with lower F1 scores?

Generalizability: How does the model perform on an entirely different dataset or in real-world applications?

Transformer Layer Optimization: Given that the focus of this research is on Triple Fusion models and Transformer layers, further research could delve deeper into the granularity of Transformer configurations.

Regularization Techniques: How effective would regularization techniques like dropout, batch normalization, etc., be in mitigating the overfitting observed in some of the configurations?

By addressing these new questions and focusing on the mentioned areas, future research can further enhance the utility and performance of these machine learning architectures.

5. Conclusion

5.1. Summary of Research Findings

The central focus of this research is to assess the effectiveness of different machine learning frameworks in terms of performance metrics, including EfficientNetB0, DenseNet121, and Inception V3, as well as an ensemble technique known as Triple Model Fusion both with and without a Transformer layer. Among the tested configurations, Triple Model Fusion emerged as the most potent, achieving an accuracy rate of 65% and macro- and weighted-average F1 scores of 0.68 and 0.65, respectively.

5.2. Interpretation of Conclusions

The empirical results substantiate that the Triple Model Fusion architecture has a marginal advantage in terms of both accuracy and F1 score metrics, making it a relatively robust choice for classification tasks. Interestingly, the inclusion of a Transformer layer did not significantly enhance the model's performance but rather showed a slight decline, particularly in macro-averaged F1 score. This indicates that while the Transformer layer increases computational complexity, its contribution to model performance is minimal.

References

[1] S. Ritz-Timme *et al.*, "Age estimation: the state of the art in relation to the specific demands of forensic practise," *Int. J. Legal Med.,* vol. 113, pp. 129-136, 2000. DOI: 10.1007/s004140050283

[2] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv:1810.04805*, vol. preprint October 11. Available: https://doi.org/10.48550/arXiv.1810.04805

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn. (CVPR)*, Las Vegas, Nevada, 2016, pp. 770-778: IEEE.

[4] A. Vaswani *et al.*, "Attention is all you need," in *Proc. 31st Conf. Neural Inform. Process. Syst. (NIPS 2017)*, Long Beach, CA, USA, 2017, vol. 30: NeurIPS.

[5] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn. (CVPR)*, Boston, Massachusetts, 2015, pp. 1-9: IEEE.

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR 2015)*, San Diego, 2014: Computational and Biological Learning Society (CBLS).

[7] S. Qiu. (n.d., August 23, 2023). *UTKFace large scale face dataset*. Available: https://susanqq.github.io/UTKFace/

[8] Skillcate. (n.d., August 24, 2023). *Age detection model using CNN: A complete guide*. Available: https://medium.com/@skillcate/age-detection-model-using-cnn-a-complete-guide-7b10ad717c60

[9] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," in *Proc. 12th USENIX Symp. Operat. Syst. Design Implement. (OSDI '16)*, Savannah, GA, USA, 2016, pp. 265-283: USENIX.

[10] X. Zhang, "Attention model ensemble method for cataract diagnosis," master's thesis, Sichuan University, Chengdu, 2021.

[11] J. Alammar. (2018, August 26, 2023). *The illustrated transformer*. Available: https://jalammar.github.io/illustrated-transformer/