Project Report - ECE 176

Nicole Li
ECE
A17344793

Farhat Ahmed
ECE
A17359822

Abstract

This project aims to develop a system capable of accurately predicting a person's age from facial images by identifying distinctive features. We utilize the Age, Gender, and Ethnicity Face Data CSV dataset from Kaggle [1], which has faces across different ages. Our approach began with Logistic Regression as a baseline model, followed by experiments with CNN architectures—including VGG, ResNet, and Inception Modules—for improved classification performance. While the baseline model achieved an accuracy of 0.11, our top-performing CNN model with reached an accuracy of 0.3810. These results show the effectiveness of CNNs in age detection, which could be largely applied in personalized marketing and human-computer interaction.

1 Introduction

1.1 Problem Description

The motivation for this project comes from the growing need for automated systems across various real-world applications. An accurate age detection system can play a crucial role in multiple areas. For examples, it can enforce legal restrictions to make sure that people under 21 do not purchase alcohol. At the same time, it can also provide preferential services for the elderly, such as priority queues or discounted ticket prices. Moreover, by mitigating the risk of fraudulent age declarations through fake IDs, such a system can help uphold the law. Beyond regulatory applications, precise age estimation can enhance human-computer interaction by allowing adaptive interfaces tailored to users' age-related needs, and also support targeted marketing strategies by promoting products that appeal to specific age groups.

1.2 Understanding the Problem

Age estimation involves detecting subtle and progressive changes in facial features that occur as a person ages. Age detection requires a system to perform analysis of small details such as wrinkle patterns, skin smoothness, and changes in facial contours. These details can be affected by a range of external factors, making it essential for the model to be both robust and adaptive. Moreover, the inherent diversity of human faces demands a system that generalizes well across different ethnic backgrounds and environmental conditions.

1.3 Proposed Approach

Using a dataset from Kaggle where the labels were discrete age values and the inputs were 48x48 grayscale images of faces, we preprocessed the data so that age labels were transformed into ranges of 5 years instead of single discrete values to facilitate classification. We started off with a simple Logistic Regression Model to obtain baseline results for classification on the dataset. Next, we experimented with multiple convolutional neural network architectures and compared their accuracy to select the most optimal one. We tested the dataset with VGG-16, DenseNet, ResNet, and Inception.

For each model, we performed hyperparameter tuning on parameters like the learning rate, weight decay, and epochs to achieve the best result with each model.

1.4 Summary of Results

Of all our models, the Logistic Model predictably displayed the least accuracy with a high of 0.11 since the model is too simple to capture the complexity of face image input. The VGG-16 Model displayed a final train accuracy of 0.5625 a final validation accuracy of 0.3692, and a final test accuracy of 0.3810. DenseNet-121 achieved a final train accuracy of 0.5, a final validation accuracy of 0.3299, and a final test accuracy of 0.3285. Moreover, the ResNet model reached a final train accuracy of 0.5312, a final validation accuracy of 0.3372, and a final test accuracy of 0.3259. Our final model, the Inception Model achieved a final train accuracy of 0.625, a final validation accuracy of 0.3625, and final test accuracy of 0.3715.

The underwhelming performance of the DenseNet may result from it being to complex for the dataset. Of all the models experimented, the Inception model displayed the highest train accuracy. However, the test accuracy was 25 percentage points lower than the train accuracy suggesting some degree of overfitting. Furthermore, the VGG-16 model showed the highest test accuracy of 0.3810 which is close to its training and validation accuracies, indicating that the model is not overfitting. Overall, the Inception Model proved the most optimal in its training accuracy while the VGG-16 proved the most optimal in generalizability.

2 Related Work

One key architecture that inspires our model is the Inception model as described in the paper Going deeper with convolutions [4]. In this work, the authors propose the inception model for its improved use of computational resources, achieved by a design that increases both the depth and width of the network while keeping the computational budget constant.

Another related work for our project is presented in Deep Residual Learning for Image Recognition [2]. This work shows how training very deep networks can be made easier by having the layers learn residual functions. In this way, the layers can learn the changes needed rather than the full transformation.

The paper Densely Connected Convolutional Networks [3] describes the advantages of DenseNets over other networks by performing image recognition tasks on the CIFAR-10, CIFAR-100, SVHN, and ImageNet datasets. Unlike ResNet which combines layers by summing them, DenseNet combines layers through concatenation. The paper finds that DenseNets display some advantages such as addressing the vanishing gradient problem, strengthening feature propagation, encouraging feature reuse, and reducing the number of parameters.

3 Method

3.1 Logistic Regression Model

We started off with a simple Logistic Regression model to obtain some baseline results for the dataset. We chose a Logistic Regression model instead of Linear Regression because we are performing classification instead of regression since the 24 age labels for the images are ranges instead of discrete values. The training of this model is performed using the Adam optimizer with a learning rate of 0.000001 and a weight decay of 0.0001 over 10 epochs. These hyperparameters were found through tuning over a reasonable range of values for these parameters.

3.2 VGG Structured Model

Our VGG model follows the structure of the VGG-16 Model. The model includes 5 convolutional blocks with two convolutional layers each with a BatchNorm layer and Relu layer. All convolutional blocks are interspersed with a max pooling layer except for the fifth block. Then, the output feature map is flattened and passed through three fully connected layers before classification occurs for the 24 age groups.

The training of this model is performed using the Adam optimizer with a learning rate of 0.001 and a weight decay of 0.000001 over 10 epochs. These hyperparameters were found through tuning over a reasonable range of values for these parameters.

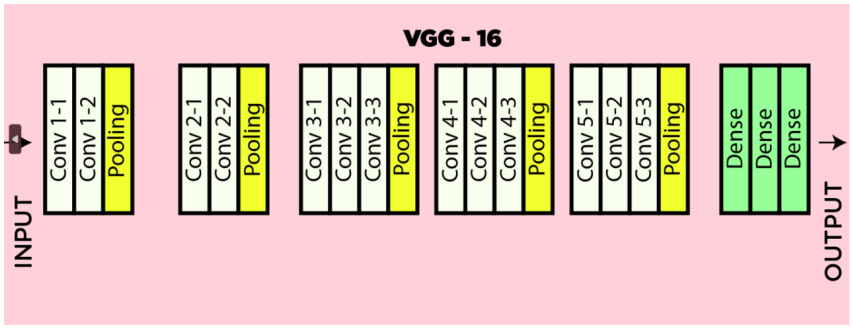

Figure 1: VGG-16 Architecture

3.3 DenseNet Structured Model

Our DenseNet-121 model is composed of Dense blocks and transition layers. Each transition layer is composed of a batch norm layer, convolutional layer, relu layer, and a max pooling layer. The DenseBlocks are made up of BottleNeck layers which are comprised of 4 convolutional layers with batch norm and relu layers in between. The inner channel of these layers is 4 times the growth rate k. The input of the BottleNeck layer is concatenated with the output. The DenseBlock combines n number of BottleNeck layers with steadily growing input channel set by the growth rate k. In the main DenseNet layer, there are four dense blocks and 3 transition layers between these blocks. The first dense block contains 6 BottleNeck layers, the second dense block contains 12, the third contains 24, and the last dense block contains 16. The output feature map of the last dense block is fed into a three fully connected layers before classification. In the process of training, some elements like the average pooling layer before the fully connected layers were removed to improve the accuracy.

The training of this model is performed using the Adam optimizer with a learning rate of 0.001 and a weight decay of 0.000001 over 10 epochs. These hyperparameters were found through tuning over a reasonable range of values for these parameters. Additionally, from hyperparameter tuning, the ideal growth rate was found to be 16.

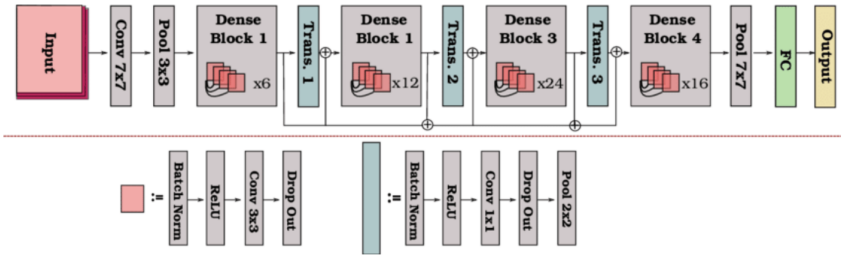

Figure 2: DenseNet-121 Architecture

3.4 ResNet Structured Model

Our ResNet model begins with a 7×7 convolutional layer followed by a max pooling layer. The core of the network consists of several residual blocks. Each block includes two sequential 3×3 convolutional layers with ReLU activations. A skip connection bypasses these convolutions, either through identity mapping or via a 1×1 convolution when the dimensions differ, which helps us preserve information and allows the training of much deeper architectures. We then use global average pooling to reduce the feature map to a fixed-size vector, which is then fed into a fully connected layer to output predictions for the 24 age classes.

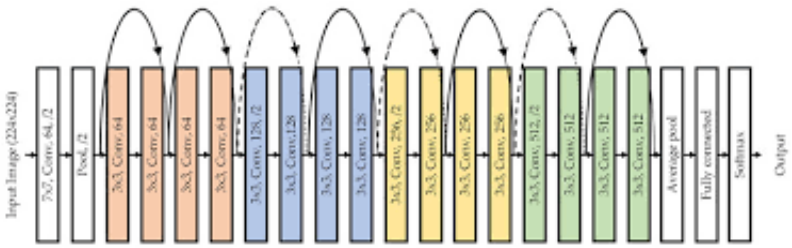

Figure 3: ResNet-18 Architecture

Training of this model is performed using the Adam optimizer with a learning rate of 1e-4, and we use the categorical crossentropy loss function over 10 epochs

3.5 Inception Structured Model

Our Inception module-based model processes the input through several parallel convolutional operations. The inception module comprises four branches: one branch applies 1×1 convolutions to capture fine details, another branch performs 3×3 convolutions to extract intermediate features, a third branch uses 5×5 convolutions to capture broader contextual information, and the fourth branch employs max pooling to retain dominant features. The outputs from these branches are concatenated, creating a rich feature map that encompasses both local details (such as wrinkles and skin texture) and global facial structure. This comprehensive feature map is then flattened and connected to a fully connected layer, which classifies the input into one of the 24 age groups.

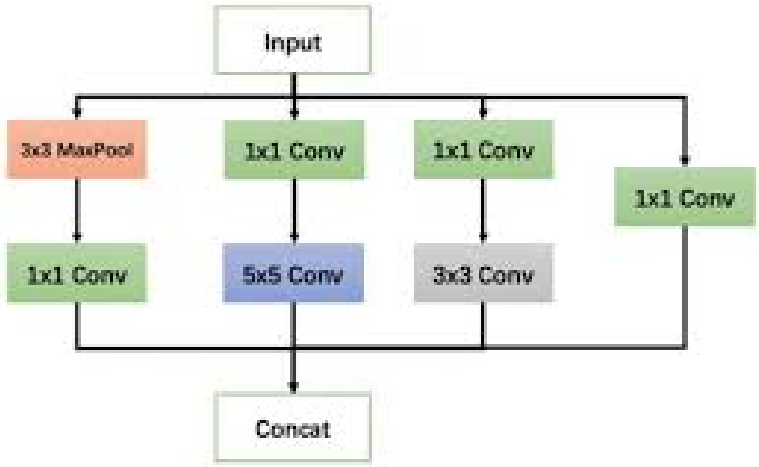

Figure 4: Inception Architecture

Training of this model is also performed using the Adam optimizer with a learning rate of 1e-4 over 10 epochs, using the categorical crossentropy loss function. Compared to other works, the use of multi-scale processing allows the network to adaptively capture features at various resolutions.

4 Experiments

4.1 Dataset

Our dataset is sourced from the Kaggle collection Age, Gender, and Ethnicity Face Data CSV [1] and comprises 23,479 facial images in grayscale. Each image is stored as a 48×48 pixel array (with a

single channel), where pixel intensity values are provided as space-separated strings.. Every image is annotated with its corresponding age. Despite the relatively low resolution, the extensive range of age groups and detailed labeling make this dataset a valuable resource for advancing robust age detection models across various applications.

4.2 Preprocessing

During the preprocessing, we validated the data to confirms that each array contains exactly 2,304 values, corresponding to the 48×48 image dimensions. These arrays are then converted into PyTorch tensors and reshaped into a single-channel format, ready for input into our neural network models. Recognizing that predicting a specific age is not feasible due to inherent challenges in variability and subtle facial changes, we divided the age labels into 5-year intervals, resulting in age groups such as "0-4", "5-9", and so on up to "115-119". This grouping gave us 24 distinct classes, which allows for a more robust and manageable classification task. Finally, we split the dataset into training, validation, and testing sets.

4.3 Results

4.3.1 Logistic Regression Model

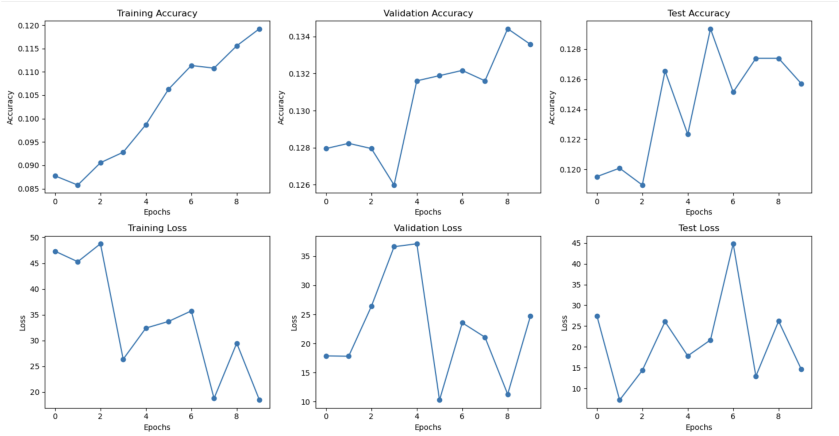

Figure 5: Training, Validation, and Test Loss and Accuracy for Logistic Model

For Logistic Regression, the training accuracy clearly rises from 0.09 to just below 0.12 and the training loss decreases over the epoch range. However, the validation accuracy also rises over the epochs but dips slightly towards the end. A similarly irregular pattern is seen in the validation loss with a rise in the loss in the middle of the graph. These results along with general low accuracy indicate that the Logistic Model does not adequately classify the age range based on the images of the dataset.

4.3.2 VGG Structured Model

For the VGG-16 Model, the train accuracy rises steadily over the 10 epochs rising to a high of 0.5625 before sharply dropping in the tenth epoch. The train loss also decreases across the epochs although with some fluctuations in between. Furthermore, more consistently than the train, the validation accuracy rises across the epochs consistently concluding with the value of 0.3692 and the validation loss also more consistently decreases across the epochs. The final test accuracy of the model is 0.3810 which is close to its training and validation accuracies indicating that there is little overfitting and the model is generalizable.

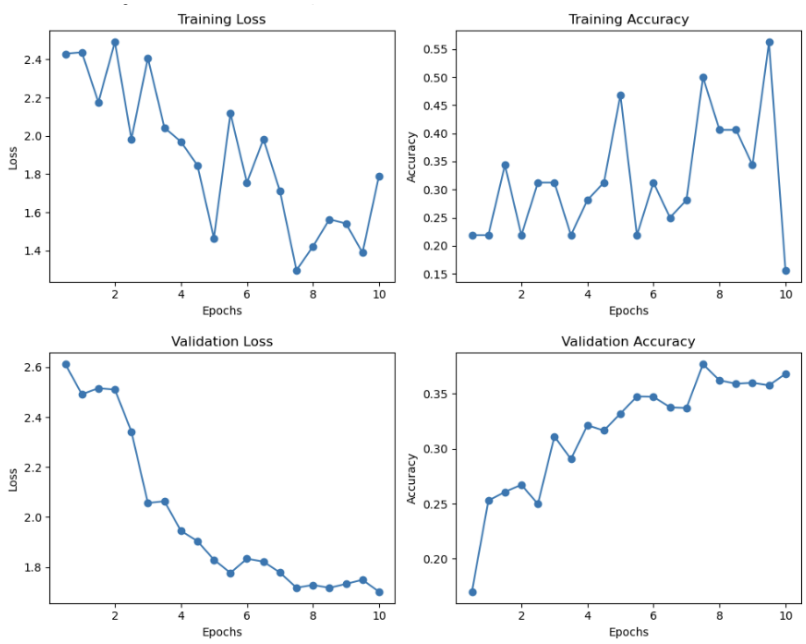

Figure 6: Training and Validation Loss and Accuracy for VGG-16 Structured Model

```
Test Loss: 1.6880, Test Accuracy: 38.10%
```

Figure 7: Test Loss and Accuracy for VGG-16 Structured Model

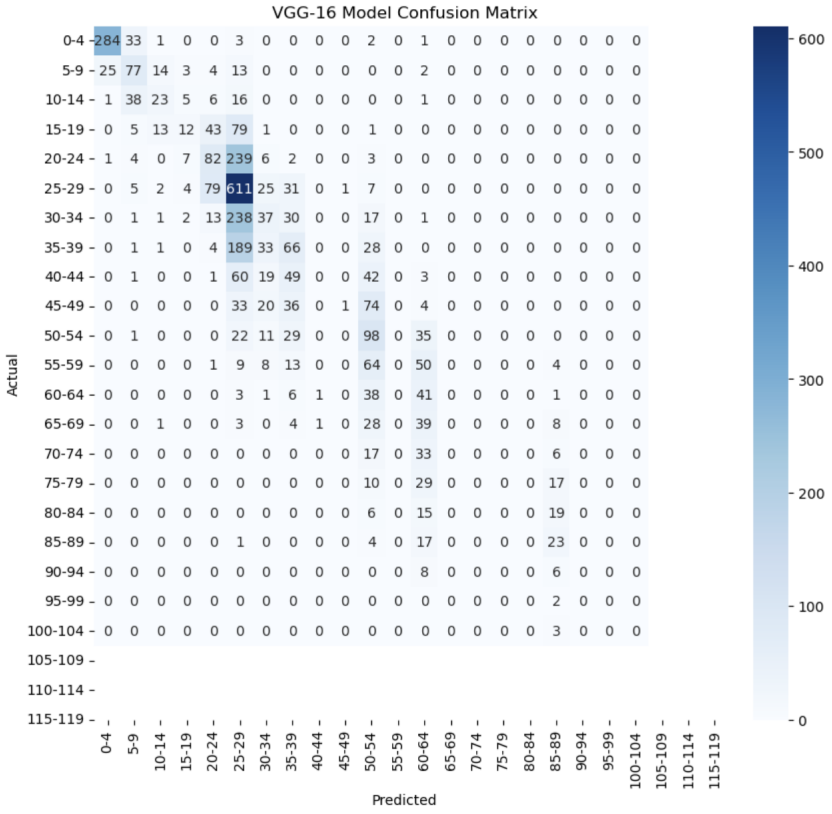

Figure 8: Confusion Matrix for VGG-16 Structured Model

The confusion matrix illustrates that VGG-16 model is performing strongly with all the values clustered along the diagonal, indicating even when the model's prediction is outside the correct age range, the error is not too large.

4.3.3 DenseNet Structured Model

For the DenseNet-121 Model, the train accuracy rises steadily over the 10 epochs to a high of 0.5 before settling to a lower value of 0.2812, however the train accuracy graph contains many fluctuations. The train loss also decreases overall across the epochs with a smaller degree of fluctuations. More consistently than the train, the validation accuracy rises steadily across the epochs reaching a final accuracy of 0.3299 and the validation loss decreases consistently over the epochs. The model achieves a final test accuracy of 0.3285 which is fairly close to its final validation and training accuracies.

The final test accuracy(0.3285) of DenseNet is substantially lower than the final test accuracy(0.3810) of the VGG-16 Model which suggests that the DenseNet model may be adding unnecessary complexity to the data with its complex structure containing many layers.

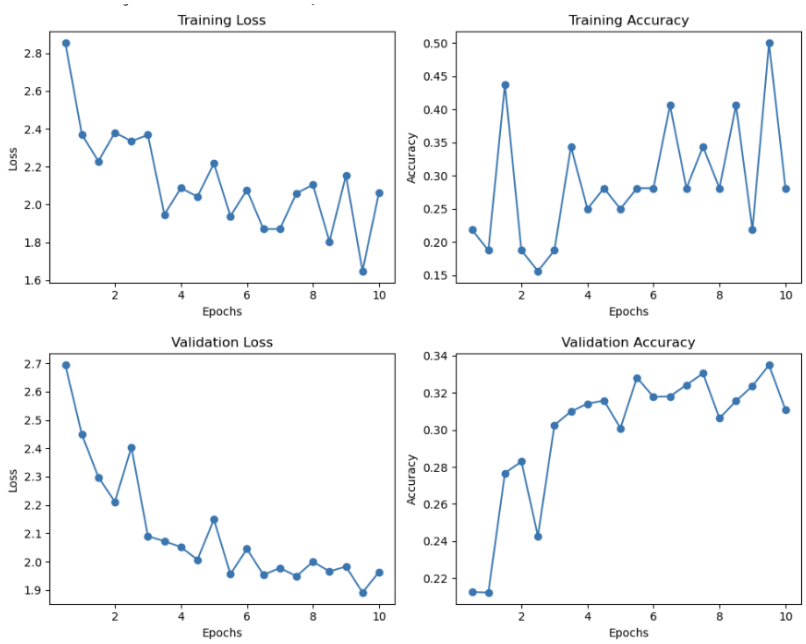

Figure 9: Training and Validation Loss and Accuracy for DenseNet-121 Structured Model

```
Test Loss: 1.9184, Test Accuracy: 32.85%
```

Figure 10: Test Loss and Accuracy for DenseNet Structured Model

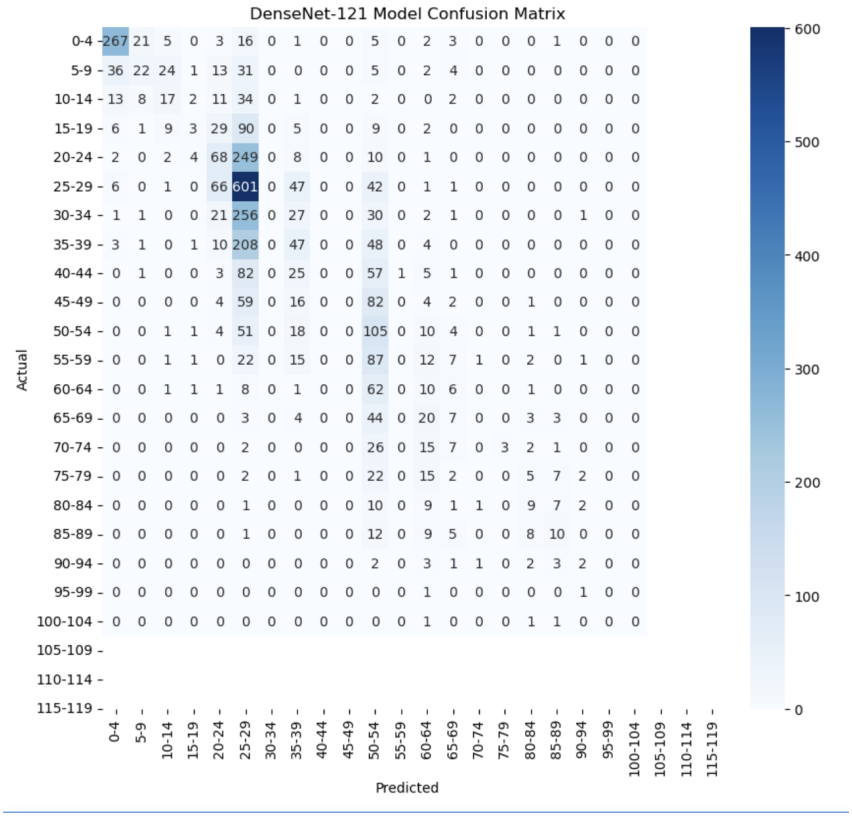

Figure 11: Confusion Matrix for DenseNet Structured Model

The confusion matrix illustrates that the Dense-121 model is performing well strongly with most values clustered along the diagonal, indicating even when the model's prediction is outside the correct age range, the error is not too large.

4.3.4 ResNet Structured Model

We implemented a ResNet-based architecture for age detection. Each residual block is made of multiple convolutional layers and we have a skip connection which bypasses these transformations and adds the original input back to the block's output. This design ensures that gradients flow more effectively through the network, preserving fine-grained details necessary for distinguishing subtle age-related features such as wrinkles and skin texture. It also captures higher-level context, like facial contours, by allowing deeper layers to learn more complex representations.

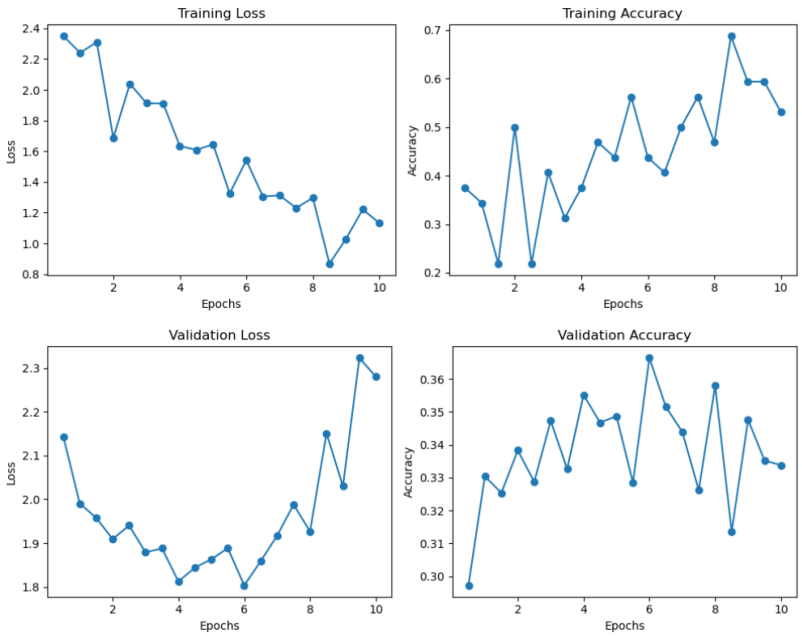

Figure 12: Training and Validation Loss and Accuracy for ResNet Structured Model

```
Test Loss: 2.3441, Test Accuracy: 32.59%
```

Figure 13: Test Loss and Accuracy for ResNet Structured Model

The ResNet model shows a clear downward trend in training loss, which shows that it is successfully learning to extract meaningful features from the data. We see that the validation loss also decreases overall, though it exhibits some fluctuations. The validation accuracy also fluctuates, which could be due to variations in batch composition, as well as the challenge of generalizing to unseen data. The overall trend still shows an improvement from the earlier epochs. By the final epoch, the model achieves a notable gap between training and validation accuracy, which may indicate some level of overfitting. The final test accuracy is similar to the final validation accuracy, which indicates that the model retains some generalization capability.

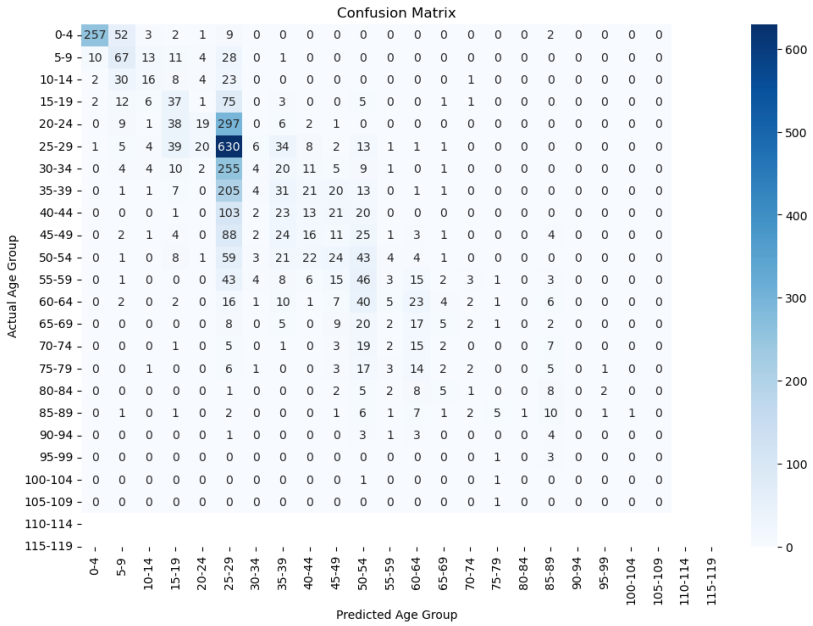

Figure 14: Confusion Matrix for ResNet-18 Structured Model

The confusion matrix shows that most entries are around the main diagonal which indicates that the model correctly classifies most samples within their respective age groups. The majority of misclassifications occur in adjacent age categories, suggesting that while the model occasionally mispredicts an age range, it rarely makes extreme errors.

4.3.5 Inception Structured Model

We implemented an inception-like structure for multi-scale feature extraction. In this design, the module incorporates several parallel branches, each performing convolution operations with different kernel sizes, such as 1x1, 3x3, and 5x5. The outputs from these branches are concatenated, forming a rich feature map that captures both fine-grained local details and broader contextual information. We found that this approach is especially beneficial for our dataset. We know that for age detection, we need to the smaller kernels to capture the small details like wrinkles, while larger kernels are essential for extracting information from larger facial areas like skin smoothness or skin contour. This flexible design enables the network to adapt to the varying scales present in the data, resulting in good performance.

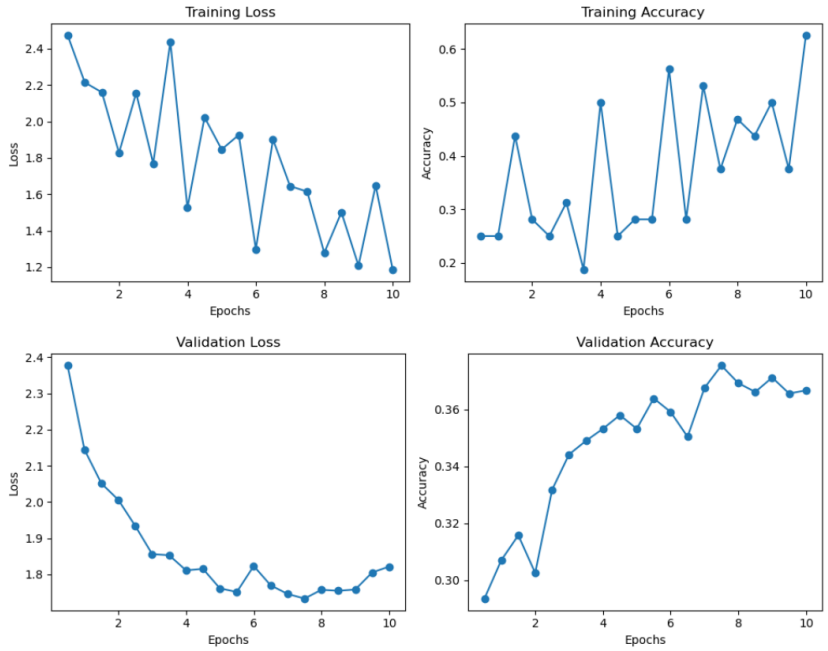

Figure 15: Training and Validation Loss and Accuracy for Inception Structured Model

```
Test Loss: 1.8179, Test Accuracy: 37.15%
```

Figure 16: Test Loss and Accuracy for Inception Structured Model

The training process shows a clear downward trend in both training and validation loss, indicating that the model is successfully learning to classify the age. While the training accuracy fluctuates, it generally trends upward, showing that the model's capacity to extract relevant features improves over time. There seems to be some overfitting happening as the training accuracy is significantly higher than the validation and test accuracy.

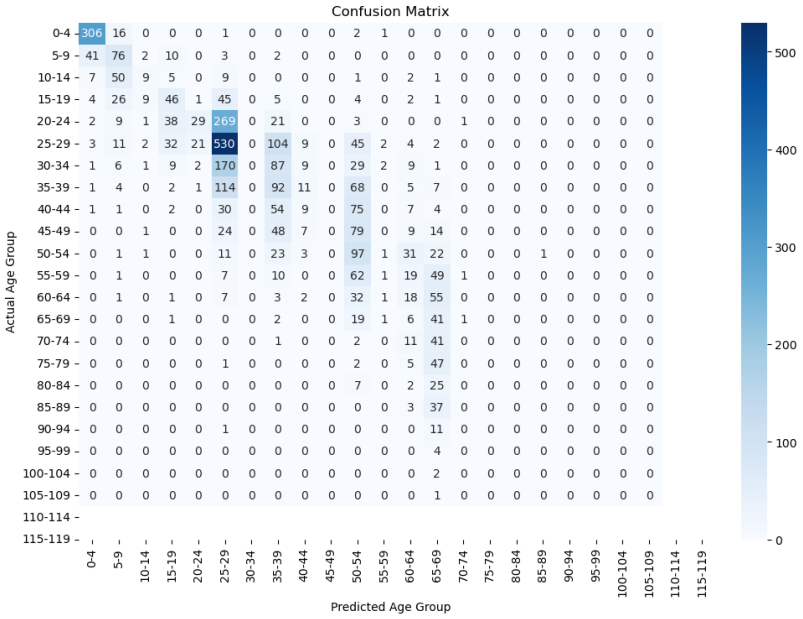

Figure 17: Confusion Matrix for Inception Structured Model

Most entries align with the main diagonal, indicating that the model accurately predicts the expected age range or a close approximation. The majority of misclassifications occur in the adjacent age group, with very few predictions at the extreme ends.

5 Supplementary Material

You should also include a video recording a presentation (with motivation, approach, results) for this project.

Github Link: Github

Video Link: Project Video

References

[1] Age, gender, and ethnicity face data csv. `https://www.kaggle.com/datasets/nipunarora8/age-gender-and-ethnicity-face-data-csv`, 2020. Accessed: 2025-03-15.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[3] Gao Huang et al. Densely connected convolutional networks. In *CVPR*, 2016.

[4] Christian Szegedy et al. Going deeper with convolutions. *CVPR*, 2014.