Robust Real-Time Age and Gender Prediction on Edge Computing devices

Muhammad Aleem Shakeel 24060005
Sheza Abbas Naqvi 26100161
Zainab Shahzad 25020238

Abstract

Real-time age and gender estimation from facial images is integral to numerous applications, from security systems to targeted marketing. However, deploying accurate and efficient models on edge computing devices remains challenging due to constraints in computational resources and latency requirements. In this work, we present a robust deep learning framework utilizing ResNet architectures with dual-output heads, simultaneously addressing age regression and gender classification tasks. Employing Bayesian inference with Monte Carlo dropout, our models quantify predictive uncertainty, thereby enhancing reliability in real-world scenarios. Using the UTKFace dataset, our discriminative and Bayesian implementations achieve competitive performance, notably attaining a gender classification F1 score of approximately 94% and an age estimation mean absolute error (MAE) as low as 5.17 years. To optimize edge deployment, the models are exported to TensorFlow.js, enabling real-time inference directly within browsers without server-side dependency. These advancements significantly enhance the practicality and scalability of demographic prediction systems on resource-constrained platforms.

1 Introduction

Age and gender estimation from facial images is a critical component in various domains such as security verification, personalized advertising, health analytics, and user authentication systems. Despite significant advancements in computer vision and machine learning, accurately predicting these demographic attributes in real-time on edge devices remains a substantial challenge due to limited computational capabilities and stringent latency requirements. Traditional methods relying on centralized servers are increasingly inadequate due to privacy concerns, network latency, and the growing need for rapid on-device inference. To address these limitations, recent research has shifted towards lightweight, efficient deep learning models capable of running directly on edge hardware.

This paper presents an advanced framework employing state-of-the-art ResNet architectures tailored for simultaneous age and gender estimation (1). Our dual-output model combines regression-based age prediction with binary classification for gender determination, optimizing accuracy while maintaining computational efficiency. Furthermore, we integrate Bayesian techniques using Monte Carlo dropout to provide predictive uncertainty measures, significantly enhancing decision-making reliability. Leveraging the comprehensive UTKFace dataset, our method demonstrates robust performance metrics, with substantial improvements in accuracy and computational efficiency suitable for edge deployment. By exporting trained models to TensorFlow.js, our solution facilitates rapid, server-independent inference directly within browsers, enhancing scalability, user privacy, and practical deployment across diverse, resource-constrained platforms.

2 Literature Review

Abu Nada et al. (2) introduced a novel form validation system that leverages deep learning to automatically detect and verify a user's age and gender from uploaded images. By linking this information to form inputs such as gender and date of birth, the system ensures data consistency and user authenticity. Their approach employs Convolutional Neural Networks (CNNs) for multi-class classification, using the Adience dataset with age categorized into eight discrete ranges to simplify the prediction task. Deployed as a RESTful web service using Flask, the system includes DNN-based face detection for feature extraction. Evaluated on a dataset of 100 student images from the University of Palestine, the model achieved 91% accuracy in gender prediction and 70% in age range classification. These results underscore the system's strong gender classification performance while highlighting the inherent challenges of age estimation due to variability in facial features. This work is highly relevant to our project as it demonstrates the practical application of lightweight CNNs for real-time demographic prediction in web forms. We aim to build upon this by deploying our own model via TensorFlow.js, enabling in-browser, webcam-based validation directly on edge devices.

Sheoran et al. (3) explored age and gender prediction using a combination of custom CNN architectures and transfer learning. Utilizing the UTKFace dataset, they compared models including VGG16, ResNet50, and SENet50, pre-trained on large-scale facial datasets like VGGFace and VGGFace2. Their findings indicate that transfer learning significantly outperforms training CNNs from scratch, especially on limited data. Among the models, ResNet50 performed particularly well, achieving a mean absolute error (MAE) of 4.65 for age prediction and 94.64% accuracy for gender classification. These results were further enhanced through architectural tuning, spatial dropout, and the Adam optimizer. This study validates our decision to use ResNet as the core architecture for our model, offering a balance between accuracy and computational efficiency. The implementation strategies presented also inform our fine-tuning and deployment plan for real-time, in-browser inference using TensorFlow.js

Huynh and Nguyen (4) proposed an advanced method for age and gender prediction, focusing on marketing and surveillance applications. Their approach utilizes a Wide ResNet architecture paired with advanced augmentation techniques—random erasing and Mixup—to boost robustness and generalization. Random erasing introduces occlusions, while Mixup generates synthetic training samples by blending pairs of images. Using the MegaAge-Asian dataset, which includes over 40,000 annotated images, their model achieved 97.8% gender classification accuracy and 82.4% age range classification accuracy. The strength of this method lies in its ability to generalize well across diverse inputs, particularly within the Asian demographic. Inspired by this, we plan to incorporate random erasing and Mixup into our training pipeline to further enhance accuracy and robustness across a broader range of facial data.

Kharchevnikova (5) focused on video-based age and gender recognition, particularly in retail environments aimed at contextual advertising. This research highlights the limitations of static predictions and proposes classifier fusion to improve accuracy in real-time video streams. Deep CNNs were trained on datasets like IJB-A, Indian Movie, EURECOM Kinect, and EmotiW2018. The system aggregates predictions across multiple frames using fusion strategies such as voting and posterior averaging. Preprocessing steps included face detection using the Viola-Jones algorithm and normalization for consistency. Implemented on platforms like PyCharm and Android, the system demonstrated high recognition reliability, even under challenging conditions such as poor lighting or suboptimal camera angles. The use of classifier fusion aligns with our goal of building a robust system that maintains high accuracy across varied real-time inputs. These strategies will inform our implementation of multi-frame prediction and preprocessing for edge deployment.

Finally, Debgupta et al.(6) proposed an improved age and gender prediction model using a Wide ResNet architecture, emphasizing its practical relevance in domains such as social media and retail analytics. Their model significantly outperforms traditional CNNs like VGG-16, achieving 96.27% accuracy and low MAE values (1.73 for apparent age and 1.65 for real age) on the APPA-REAL dataset. The architecture effectively mitigates vanishing gradient issues, and the use of data augmentation techniques—such as random erasing and Mixup regularization—contributes to better generalization and reduced training times. Evaluations on large datasets like IMDb-WIKI and APPA-REAL confirmed the model's superiority, with MAE values far lower than those of VGG-16 (e.g., 4.33 vs. 6.06 for apparent age). These findings reinforce our decision to explore Wide ResNet as an alternative to standard ResNet models, especially given its strong performance and efficiency benefits in age and gender estimation.

3 Problem Formulation

The goal of this project is to design a deep learning-based system for accurately predicting a user's age and gender from facial images in real-time. The solution has optimization constraints since it is designed for edge devices. Therefore, a compact yet powerful architecture is required to ensure low latency, high availability, efficient preprocessing, and minimal server-side computation.

The primary challenges include:

- High variance in age and appearance due to varying conditions
- Significant class imbalance in age and gender distributions within the dataset.
- Balancing model complexity with real-time performance constraints on edge hardware.

To address these challenges, we propose the following approach:

- Formulate a dual-head learning model (7) i.e. Employ age prediction as a regression task, minimizing the mean absolute error (MAE), and treat gender classification as a binary task.
- To support deployment efficiency and enable quick real-time response, it is essential that both age and gender predictions are performed using a single unifier model, maintaining the baseline results.

Our approach is validated using the UTKFace dataset. The resulting model is deployed through TensorFlow.js to enable in-browser inference.

4 Dataset

We have used UTKFace dataset (8) for training the model. The details of the dataset are shown in 1

Table 1: UTKFace Dataset, mutually exclusive for each attribute

Attribute	**Details**
Total Images	Over 20,000 facial images
Annotations	Each image is labeled with age, gender, and ethnicity
Age Range	0 to 116 years
Gender Labels	0: Male, 1: Female
Ethnicity Labels	0: White, 1: Black, 2: Asian, 3: Indian, 4: Others
Image Format	JPG images, cropped and aligned

To understand the dataset, we visualized gender and age distributions using bar plots and histograms. The dataset shows imbalanced gender split and a wide age range, grouped into six categories: Child (0–12), Teen (13–18), Young Adult (19–30), Adult (31–50), Middle Aged (51–70), and Senior (71+) (9).

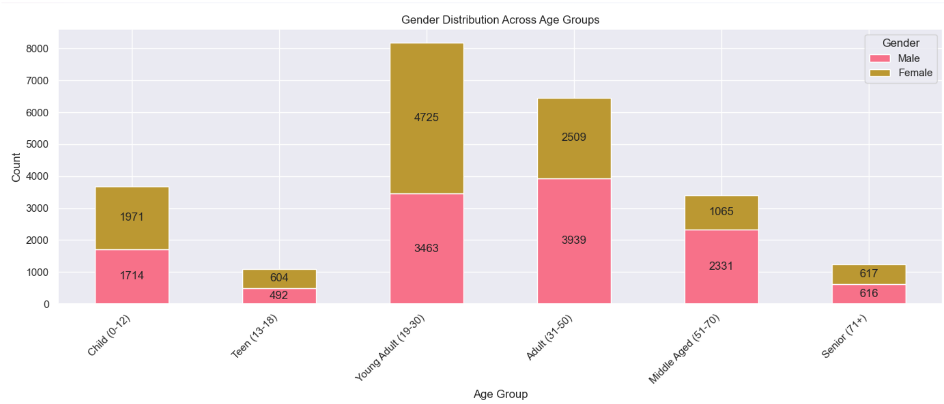

Figure 1: Class Imbalance for UTKFace Dataset

To address class imbalance in the dataset, we applied targeted data augmentation—using techniques such as random flips, brightness, contrast, saturation adjustments, and rotations. These augmentations were applied more heavily to underrepresented classes, while well-represented classes received minimal augmentation to maintain data integrity.

5 Methodology

This section gives detail overview of data processing, modelling, model training strategies, testing and deployment strategies.

5.1 Data Preprocessing

RGB facial images are preprocessed using the following steps:

1. **Stratified Sampling by Age:** We perform stratified sampling on discretized age bins to maintain age distribution across train, validation, and test splits.
2. **Augmentation:** Training images are randomly augmented with:
   - Horizontal flips
   - Brightness and contrast adjustments
   - Random rotations
   - Color saturation adjustments
3. **Normalization:** Each pixel intensity x is normalized using:
$$z = \frac{x - \mu}{\sigma}$$
   where μ and σ are the dataset mean and standard deviation.
4. **Resizing:** All images are resized to $224 \times 224 \times 3$.

We trained both deterministic and Bayesian models to evaluate the trade-offs between predictive accuracy and uncertainty estimation. While deterministic models offer faster inference and simpler training, Bayesian models enable quantification of uncertainty, making them more reliable for real-world, safety-critical edge applications.

5.2 Modelling - Deterministic Models

We implemented a deterministic deep learning framework for real-time age estimation (regression) and gender classification (binary classification) from facial images. Our approach uses ResNet-based convolutional neural networks (ResNet-34 and ResNet-152) trained via multi-task learning (10) to jointly optimize both tasks. This section describes the data preprocessing steps, model architecture, training procedure, and deployment strategy.

5.2.1 Architecture

Two backbone networks, ResNet-34 and ResNet-152, implemented in TensorFlow / Keras, extract features, followed by task-specific heads:

- **ResNet-34:** A custom implementation using Basic Blocks and residual skip connections.
  - **Age Head:** Two dense layers ($512 \rightarrow 256 \rightarrow 1$) with ReLU activations and dropout rates 0.5 and 0.3 respectively. Outputs a scalar predicted age.
  - **Gender Head:** A classification head mirroring the age head structure ($512 \rightarrow 256 \rightarrow 1$) with ReLU and dropout, followed by a sigmoid activation to predict the probability of gender. (11)
- **ResNet-152:** A deeper model fine-tuned for the dual-task objective. It provides enhanced representational power, though with increased computational demands.
  - **Age Head:** Comprises a single fully connected layer with 512 units, ReLU activation, and L2 regularization, followed by dropout (0.3), and a final linear layer producing a single scalar for age regression.
  - **Gender Head:** Contains one dense layer with 512 units and L2 regularization, followed by batch normalization, ReLU activation, dropout (0.3), and a final linear layer outputting one value. A sigmoid activation is applied downstream for binary gender classification.

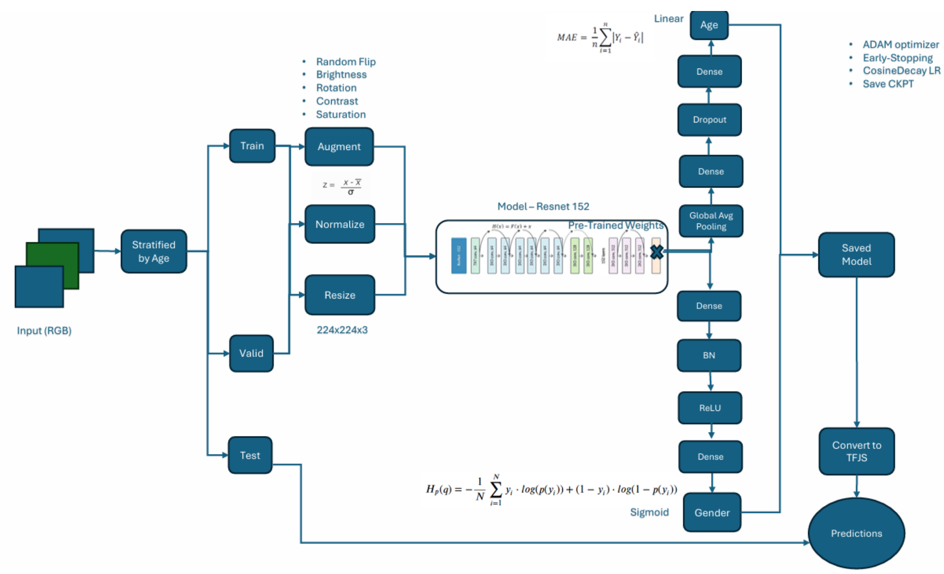

Figure 2: Architecture comparison of discriminative models using ResNet-34 and ResNet-152 backbones. Both models share identical preprocessing pipelines, with adjustments made to the bottleneck layers to align with their respective depths and capacities.

5.2.2 Loss functions

Age is a scalar regression task minimizing Mean Absolute Error (MAE):

$$L_{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

where y_i is the true age and $\hat{y}_i$ is the predicted age.

Gender prediction is treated as a Binary classification with sigmoid output, minimizing Binary Cross Entropy (BCE):

$$L_{BCE} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)]$$

where $y_i \in \{0, 1\}$ is the gender label and p_i is the predicted probability.

The model jointly learns both tasks using a weighted loss:

$$L_{total} = \lambda_{age} \cdot L_{MAE} + \lambda_{gender} \cdot L_{BCE}$$

with empirically chosen weights: $\lambda_{age} = 0.7$, $\lambda_{gender} = 1.0$. This balances performance between regression and classification objectives.

5.2.3 Training Strategy

The training strategies used for the training are:

- **Optimizer:** Adam optimizer with an initial learning rate of 1×10^{-4}.
- **Learning Rate Scheduling:**
  - ResNet-152: CosineDecay learning rate scheduler.
  - ResNet-34: ReduceLROnPlateau (factor=0.2, patience=3).
- **Regularization:** Dropout and early stopping (patience=5) are used to mitigate overfitting.
- **Checkpointing:** Models are checkpointed based on best validation loss.
- **Training Details:** Batch size = 32, Epochs = 50 (with early stopping).

5.3 Modelling - Bayesian Model

We propose a Bayesian deep learning framework for joint age estimation (regression) and gender classification (binary classification) from facial images in real time. Our methodology leverages ResNet-based convolutional neural networks (ResNet-152 and ResNet-34) with Monte Carlo (MC) Dropout to enable uncertainty quantification. This section outlines the data preprocessing pipeline, model architecture, training procedure, and inference strategy.

5.3.1 Architecture

We employ pre-trained ResNet-152 and ResNet-34 backbones for feature extraction. These are followed by two task-specific heads:

- **Age Head:** A regression head that predicts both the mean μ and variance σ^2 of the age.
- **Gender Head:** A binary classification head with a sigmoid activation.

Dropout layers are inserted after dense layers in both heads to support Bayesian inference.

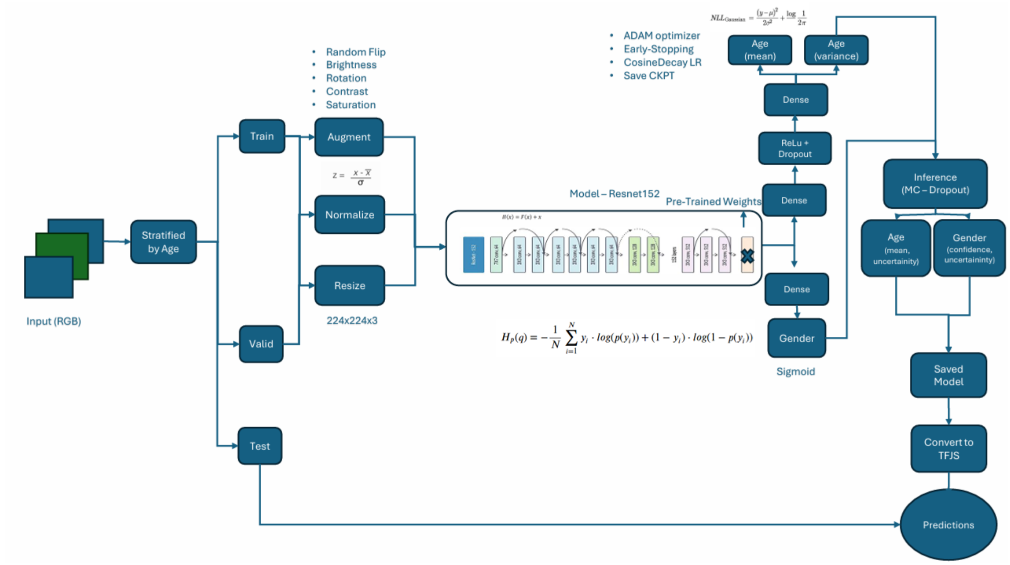

Figure 3: Architecture comparison of bayesian models using ResNet-34 and ResNet-152 backbones. Both models share identical preprocessing pipelines, with adjustments made to the bottleneck layers to align with their respective depths and capacities.

5.3.2 Loss functions

We model age as a heteroscedastic Gaussian distribution, where both μ and σ^2 are learned. The model is trained using the Gaussian Negative Log Likelihood (NLL):

$$\mathcal{L}_{NLL} = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{(y_i - \mu_i)^2}{2\sigma_i^2} + \frac{1}{2} \log(2\pi\sigma_i^2) \right)$$

Here, y_i is the ground truth age, and μ_i, σ_i^2 are the predicted parameters.

Gender is predicted using a sigmoid-activated output neuron. The Binary Cross Entropy (BCE) loss is minimized:

$$\mathcal{L}_{BCE} = -\frac{1}{N} \sum_{i=1}^{N} [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)]$$

where $y_i \in \{0, 1\}$ is the ground truth gender label, and p_i is the predicted probability. We used no weighted average in Bayesian modelling.

5.3.3 Uncertainty Estimation via Monte Carlo Dropout

To estimate uncertainty, we apply Monte Carlo Dropout during inference. We perform T stochastic forward passes through the model (12), obtaining predictive distributions:

$$\hat{y} = \frac{1}{T} \sum_{t=1}^{T} f_\theta^{(t)}(x)$$

Here, $\hat{y}$ represents the predictive mean obtained through Monte Carlo Dropout. The model $f_\theta^{(t)}(x)$ is evaluated T times with random dropout masks applied during each forward pass. By averaging the outputs, we estimate the predictive distribution, enabling uncertainty quantification. A dropout rate of $p = 0.2$ is used during both training and inference to simulate Bayesian approximation.

- **For Age:** Predictive uncertainty is computed as the variance across predicted means μ_t over T passes.
- **For Gender:** Epistemic uncertainty is quantified using the variance of the sigmoid probabilities.

This yields confidence intervals for age and uncertainty scores for gender predictions.

5.3.4 Training Strategy

- **Optimizer:** Adam optimizer is used. ResNet-152 uses CosineDecayLR, and ResNet-34 uses ReduceLROnPlateau for learning rate scheduling.
- **Regularization:** Dropout and early stopping are employed to prevent overfitting.
- **Checkpointing:** Best-performing models are saved based on validation loss.

6 Results

In this section, we present the performance evaluation of our proposed models on the joint tasks of age estimation and gender classification. We experimented with two architectures—**ResNet-34** and **ResNet-152**—under both *discriminative* and *Bayesian* settings. The models were trained and evaluated on a comprehensive dataset annotated for both age and gender attributes.

6.1 Age Estimation

Table 2 shows the Mean Absolute Error (MAE) for age prediction. Among all configurations, the discriminative **ResNet-34** achieved the lowest MAE of **5.1699**, followed closely by **ResNet-152 (Discriminative)** with an MAE of **5.26**. The Bayesian models yielded higher MAEs, with **ResNet-34 (Bayesian)** performing the worst at **7.83**, and **ResNet-152 (Bayesian)** achieving a relatively better score of **5.83**.

6.2 Gender Classification

The gender classification task was evaluated using F1 scores for both classes (male and female), as well as confusion matrices and accuracy trends.

F1 Score: As shown in Table 2, the highest F1 score for male classification was achieved by both **ResNet-152 (Discriminative)** and **ResNet-152 (Bayesian)**, each with an F1 score of **0.9370**. For female classification, the **ResNet-152 (Discriminative)** model outperformed others with an F1 score of **0.9297**, closely followed by **ResNet-152 (Bayesian)** at **0.9300**.

Confusion Matrices: Figure 4 illustrates the confusion matrices for the Bayesian models. **ResNet-152 (Bayesian)** demonstrated superior classification consistency, with 1793 true positives for class 0 (male) and 1580 for class 1 (female), and lower misclassification counts (91 and 147, respectively), compared to **ResNet-34 (Bayesian)**.

Table 2: Model Performance Comparison on Age and Gender Tasks

Model	**Age (MAE)**	**Gender (F1) - Male**	**Gender (F1) - Female**
ResNet-34 (Discriminative)	**5.1699**	0.8772	0.8700
ResNet-152 (Discriminative)	5.26	**0.9370**	**0.9297**
ResNet-34 (Bayesian)	7.83	0.9272	0.9232
ResNet-152 (Bayesian)	5.83	**0.9378**	0.9300

Training Dynamics: Loss and accuracy plots In Figure 6 & 5) further affirm the superior convergence and generalization of deeper architectures. The **ResNet-152 (Bayesian)** model exhibited smooth loss reduction and continuous accuracy improvement over epochs, indicating more stable learning behavior compared to the shallower **ResNet-34** counterpart.

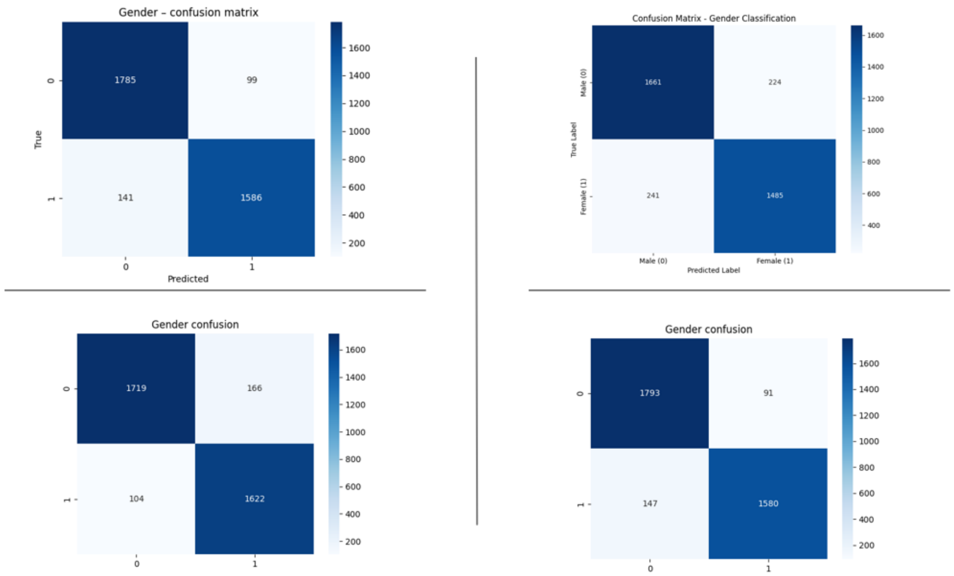

Figure 4: Confusion matrices for different model configurations: (Top-left) ResNet-152 Deterministic Model, (Top-right) ResNet-34 Deterministic Model, (Bottom-left) ResNet-34 Bayesian Model, (Bottom-right) ResNet-152 Bayesian Model.

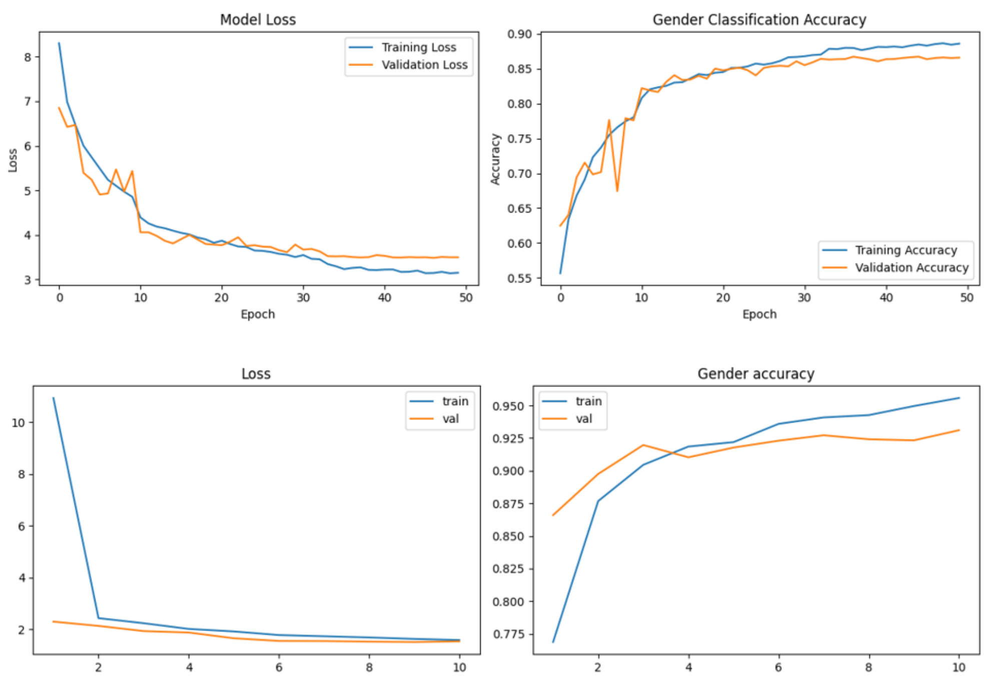

Figure 5: Training loss and gender classification accuracy curves for ResNet-34 models: (top) Discriminative model, (bottom) Bayesian model.

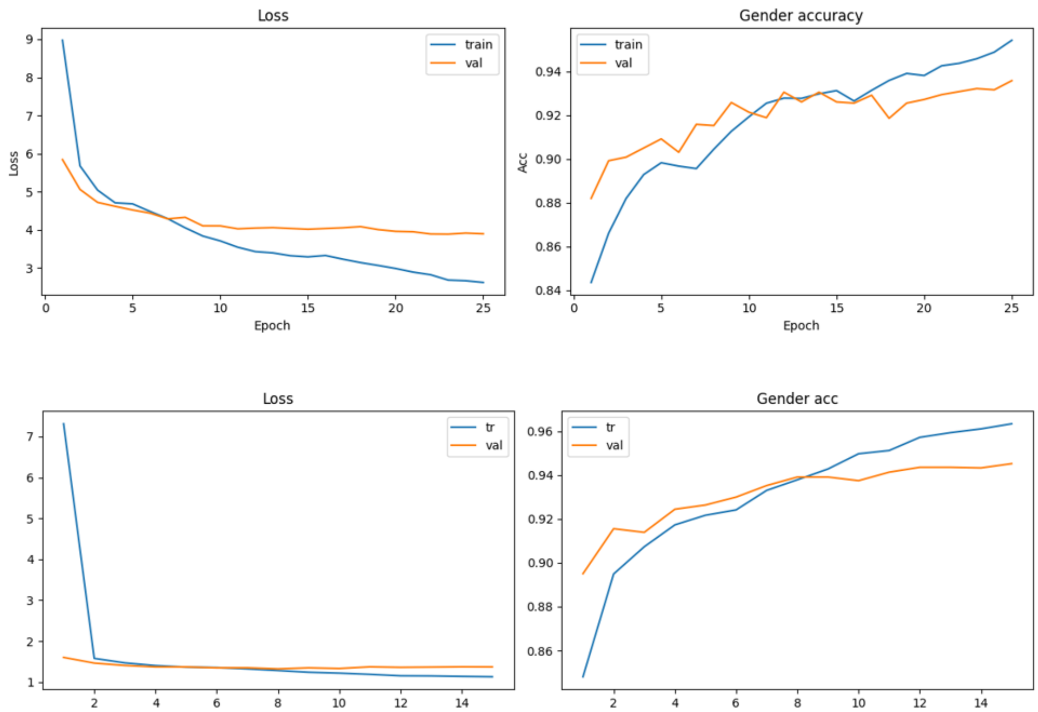

Figure 6: Training loss and gender classification accuracy curves for ResNet-152 models: (top) Discriminative model, (bottom) Bayesian model.

7 Deployment

The trained models are exported to TensorFlow.js (13) for real-time inference in browsers using webcam input. This allows on-device predictions with low latency and no server dependency, making the solution scalable and privacy-preserving. This enables:

- Low-latency, on-device predictions
- No server-side processing
- Improved user privacy and deployment scalability

The same ResNet-34 and ResNet-152 backbones are preserved in the TensorFlow.js environment, enabling consistent performance between offline training and online execution.

8 Testing on Front-End

For testing, we developed a React-based frontend interface that interacts with the TensorFlow.js models directly in the browser. During runtime, the exported models (in .json and .bin format) are loaded and stored in the cache memory of the client device to support repeated inference without reloading. This approach allowed us to evaluate all deployed models efficiently through the browser, confirming their usability, performance, and responsiveness under real-world conditions without server intervention. The results are shown in Table 3

9 Observations & Suggestions

Based on experimental evaluation, several key observations were noted:

Table 3: Latency and Processing Time Comparison Across Devices for ResNet-34 and ResNet-152

Device	Version	ResNet-34		ResNet-152	
		Latency Time	Processing Time	Latency Time	Processing Time
Laptop	i7 11th Gen	2.50 min	3.18 sec	8.78 min	3.576 sec
	Apple M1 Air 8GB	6.29 min	3.96 sec	9.06 min	4.068 sec
Phone	POCO F3 8GB (Snapdragon 870 Gaming)	2.28 min	1.98 sec	6.30 min	2.04 sec
	iPhone 14 Pro Max	7.39 min	6.66 sec	15.13 min	5.28 sec
	iPhone 13	12.20 min	7.20 sec	16.60 min	7.68 sec
Tablet	Xiaomi Tab 6	6.44 min	3.762 sec	8.97 min	3.384 sec

**The latency time highly depends on network speed.*

- ResNet-152, though powerful, was overly complex for the dataset and required significant augmentation to avoid overfitting.
- Deployment latency is highly sensitive to the user's network speed.
- Converting models from Python (float32) to TensorFlow.js (float16) introduces an accuracy trade-off shown in Table 4, particularly for regression tasks. The accuracy drop is primaraly due to compression of model during conversion. Models trained in Python use 32-bit floats. Post-conversion for web deployment, TensorFlow.js uses 16-bit floats.

Table 4: Impact of model conversion on different use cases

Use Case	Impact of float16
Image classification	Minor (1–2% drop)
Regression (MAE-sensitive)	Significant (up to 10–20% MAE increase)
Large/Deep models (e.g., ResNet152)	More sensitive to precision changes

To further enhance the performance of the deployed system, the following improvements are recommended:

- Use an extended dataset beyond UTKFace to improve generalization and demographic coverage.
- Prefer TensorFlow.js for lightweight models (e.g., MobileNet) where fast inference and small memory footprint are essential.
- For deeper networks such as ResNet-152, consider deploying using TensorFlow Lite instead of TensorFlow.js to preserve float32 precision and avoid performance degradation.
- Expand the dataset to include images under varying lighting conditions to improve the model's robustness to illumination changes.
- For models too large to deploy directly on low-end devices, apply quantization techniques to reduce model size (14) and improve efficiency without severely compromising accuracy

10 Conclusion

In this project, we successfully developed a robust, real-time age and gender prediction system leveraging deep learning and deployed it for efficient in-browser inference using TensorFlow.js. By training dual-headed models on the UTKFace dataset, we simultaneously achieved gender classification and age estimation with high accuracy. Rigorous preprocessing and data augmentation techniques contributed to balanced training and improved generalization. Our model attained a gender classification accuracy of around 94% and a mean absolute error (MAE) of 5.17 for age prediction, indicating strong performance across diverse demographic profiles. To ensure deployment-readiness, we tested the system through a React-based frontend, validating the model's responsiveness, inference latency, and stability directly within the browser environment. For future work, we recommend improving demographic fairness and accuracy by extending the training dataset to include a broader range of ethnicities and age groups. Additionally, for better deployment efficiency on low-resource devices, adopting lightweight architectures such as MobileNet is advisable to maintain performance while reducing computational overhead.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[2] A. M. A. Nada, E. Alajrami, A. A. Al-Saqqa, and S. S. Abu-Naser, "Age and gender prediction and validation through single user images using cnn," *International Journal of Academic Engineering Research*, vol. 4, no. 8, pp. 21–24, 2020.

[3] V. Sheoran, S. Joshi, and T. R. Bhayani, "Age and gender prediction using deep cnns and transfer learning," in *Computer Vision and Image Processing*. Springer Singapore, 2021, pp. 345–356.

[4] H. T. Huynh and H. Nguyen, "Joint age estimation and gender classification of asian faces using wide resnet," *SN Computer Science*, 2020.

[5] A. S. Kharchevnikova and A. V. Savchenko, "Video-based age and gender recognition in mobile applications," in *Proceedings of the IV International Conference on Information Technology and Nanotechnology (ITNT-2018)*, ser. CEUR Workshop Proceedings, vol. 2210. CEUR-WS.org, 2018.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[7] E. Voita, D. Talbot, F. Moiseev, R. Sennrich, and I. Titov, "Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 5797–5808.

[8] Z. Zhang, "Utkface dataset," https://susanqq.github.io/UTKFace/, accessed: 2024-05-18.

[9] ——, "Utkface: A large-scale facial dataset for age, gender, and ethnicity prediction," in *Workshop on Computer Vision and Pattern Recognition (CVPR)*, 2017, accessed: 2024-05-18. [Online]. Available: https://susanqq.github.io/UTKFace/

[10] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv preprint arXiv:1706.05098*, 2017.

[11] P. Michel, O. Levy, and G. Neubig, "Are sixteen heads really better than one?" *arXiv preprint arXiv:1905.10650*, 2019.

[12] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" *Advances in neural information processing systems*, vol. 30, 2017.

[13] T. Zhao *et al.*, "Deploying deep learning models on the web with tensorflow.js: Challenges and solutions," *ACM Computing Surveys*, 2022.

[14] B. Jacob *et al.*, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2704–2713.