SMAI MINI PROJECT REPORT

Assumption.

Test data age, gender, and ethnicity distribution are identical to training data. Did not handle the case where the model is expected to predict ages greater than ages present in train data.

Loss Function

Treated age prediction as a regression problem, have experimented on MSE loss, L1 loss, and also Smooth loss.

Problem with MSE loss: as there are noisy images, there will be outliners. MSE loss is more sensitive to outliers. Here if for any image the absolute difference between the true value and predicted value is greater than 25, is considered as outliner. Have arrived at value 25 by examining the data(the standard deviation of train data is around 20) and have printed the images and checked for outliers on predicted train data, where the absolute difference is greater than 20,25,30…

Solution: Smooth L1 loss: Smooth L1 Loss provides robustness to outliers, which is beneficial in the early stages of training when predictions can be quite inaccurate. SmoothL1Loss(beta=25) here beta=25 specifies for ages less than 25 loss function is quadratic and be treated as MSE loss. For ages greater than 25 loss function is linear. By doing this model tries to be more accurate on images that are not outliners.

As the model begins to converge switches to MSE Loss as it focuses on minimizing the average squared error, thereby promoting smaller residuals.

In this way, the model is initially exposed to a more forgiving environment (with Smooth L1 Loss), and then gradually introduced to a stricter environment (with MSE).

Model selection:

Because of the generalisability of traditional machine learning algorithms have experimented with XGboost, Random Forest, SVR, and linear regression.

Extracted features from pre-trained models and used them as input. All traditional algorithms did not yield good results on test MAE.

Have tested various pre-trained models like resnet(18,34,51,101) and efficient net(bo,b4,b7) efficient net having large parameters took a lot more time for training. Resnet models gave good yield on test mae.

As the number of train images is less, adding more layers on top of pre-trained models did not yield good results. So just added on linear layer, and made the output as 1 as it is a regression problem.

Data Augmentation:
Augmentation experimented:

RandomHorizontalFlip(),

- Faces are generally symmetrical, and flipping them horizontally doesn't change the inherent age characteristics. This can help the model generalize better to new faces seen from different angles.

RandomRotation(), # Rotate +/- 10 degrees

- Small angles (like +/- 10 degrees) are useful because people don't always face the camera directly in real-life scenarios.

ColorJitter(brightness=0.2, contrast=0.2, saturation=0.2, hue=0.1),

- Modifying brightness, contrast, saturation, and hue helps the model become robust against various lighting conditions and camera qualities, which is particularly useful for age estimation as lighting can significantly affect perceived age.

RandomResizedCrop(n_px, scale=(0.8, 1.0)),

RandomPerspective(distortion_scale=0.1, p=0.5, interpolation=3)

- This can help simulate the effect of taking a photo from slightly different angles and distances, adding robustness.

RandomGrayscale(p=0.2),

- Occasionally converting images to grayscale can force the network to focus on structural features of the face rather than color, which is beneficial since age prediction should be invariant to color. This especially helps because there are people with different ethnicities in the dataset.

GaussianBlur(kernel_size=(5, 9), sigma=(0.1, 5)) # Apply Gaussian Blur with a kernel size between 5 and 9.

- Applying a mild blur simulates minor focus issues that might occur in real-world photography and helps prevent the model from learning to rely excessively on high-frequency details which might not be relevant for age prediction.

Transformations (augmentations) increase the diversity of training data. This helps the model to learn features that are invariant to such transformations, leading to better generalization.

The model first learns with augmented data to increase its robustness and, after a certain number of epochs, decided to reduce the intensity of the transformations or revert to using the original data without augmentation. This can help the model fine-tune training data, which helps if the test distribution is identical to the train.

Optimizer:

Adam combines the advantages of the **Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp).**

AdamW is a variation of the Adam algorithm that separates weight decay from the optimization steps taken based on the gradient.

Used AdamW as Adam is not converging well.

Ensemble methods:

As only half of the test data is revealed, decided to use ensemble methods. I have used various architectures with varying Augmentation techniques, and loss function taken my best submissions on test data, applied ensemble methods, and got good results. All predictions are averaged out.

Training Loop:

Performed stratified split with validation size being 10 percent. As I know the mean and standard deviation of validation data(from train data and stratified split) after every epoch they were printed to see the stability of the model along with training loss and validation loss. Also implemented early stopping with varying values of patience values.

Future work.

Exploring vision transformers for age prediction.

Use of saliency maps to detect faces in the images and training models on only faces.

Generating close images for underrepresented ages with GANS.

Exploring other ways for age detection.

Using regression has some problems associated with it.

As there is no clear distinction between ages, every individual age differently mainly decided by genetic factors. Even for humans it is very difficult to predict the exact age, humans are good at predicting the range of ages than a single age. People of different ethnicities age differently.