AI6126: Advanced Computer Vision

Ong Jia Hui (G1903467L)
JONG119@e.ntu.edu.sg

Project 1: CelebA Facial Attribute Recognition Challenge Report

1 Exploratory Data Analysis (EDA)

The objective of this challenge is to build a high performance multi-label classifier for face attributes classification. The dataset used is from CelebA [7] which contains 200 thousand cropped and aligned faces. The dataset will be split into 162,770 images for training, 19,867 images for validation and 19,962 images for testing. There are some constraints set for this challenge which include model training strictly on the train set, restricted use of ImageNet pre-trained models only and ensemble of models is not allowed. Prior to building the model, it is important to perform EDA in order to better understand the dataset given.

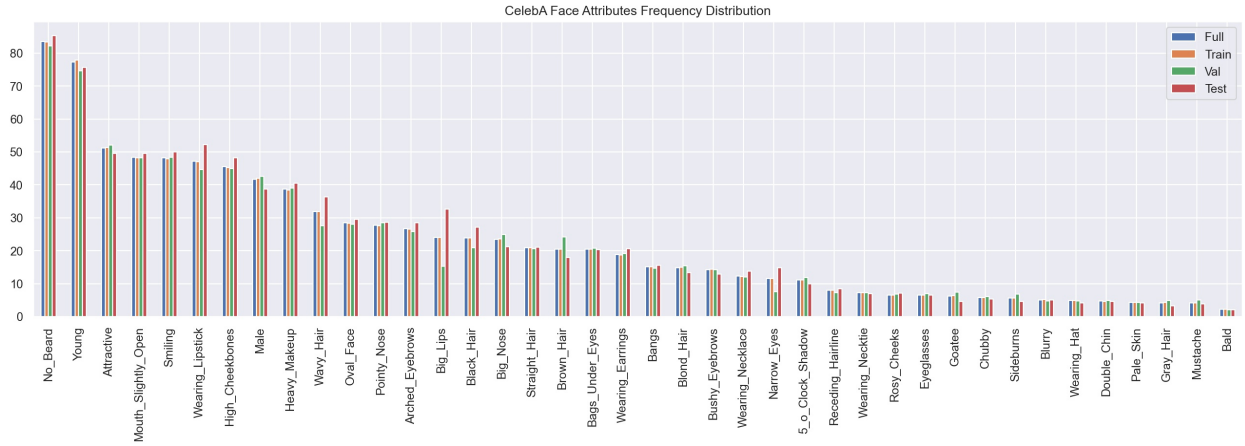


Figure 1: Frequency distribution plots of all 40 attributes for each dataset splits.

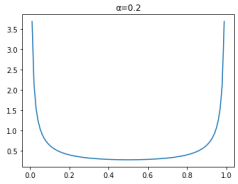
Figure 1 shows that the distribution of all dataset splits are highly similar to that of the original full dataset and the splits looked to be stratified sampled for train and validation set. For the test set, some attributes like *Wearing_Lipstick*, *Wavy_Hair* and *Big_Lips* looked to be sampled more frequently. Furthermore, the distribution plot also revealed that the full dataset is highly imbalanced with very common attributes like *No_Beard* and *Young* weighing above 70% and very uncommon attributes such as *Mustache* and *Bald* of less than 5%. This tells us that the data imbalance issue needs to be accounted for during model design. During the EDA process, multiple mislabelled images were found, some include mislabelled males as females and mislabelled women with *No_Beard* set to -1. In addition, the Celeba dataset is found to contain 131 duplicated images with 99 of them having different attribute annotations.

2 Implementation

2.1 Data Augmentation

Data augmentation is an important step in many computer vision tasks due to the scarcity of labelled dataset. It helps to "increase" the training set by creating variations of the original images. This expanded dataset alleviates the problem of overfitting and allows the model to generalize better. In this assignment, I have tried multiple data augmentation techniques recommended by various papers. Some of them include

ColorJitter [4], FancyPCA [5], random brightness and contrast [4] and adding of random Gaussian noise. All of the above were applied using a Python library called Albumentations [1]. However, they did not improve the model accuracy except for random erasing [8][15], which crops out random locations in an image to increase the difficulty of learning. Hence, for the train set, the images are center cropped to 198×158 with 50% probability of horizontal flip [4] and 50% probability for affine transformations (shift, scale, rotate) [2], before RGB normalization and conversion to tensors. For the validation and test set, the images are center cropped to 198×158 before RGB normalization and conversion to tensors.



(a) Beta distribution of $\alpha = 0.2$ for λ sampling

(b) Example of mixed up images

As recommended by He et al. in their Bag of Tricks paper [4], MixedUp training [14, 12] is an alternate data augmentation technique. In the training loop, pairs of images in a batch are interpolated using a λ value sampled from a beta distribution of $\alpha = 0.2$ (hyperparameter). The training loss is also computed using a weighted (λ) linear combination of the two losses calculated from the two labels of the mixed images.

2.2 Model Architecture

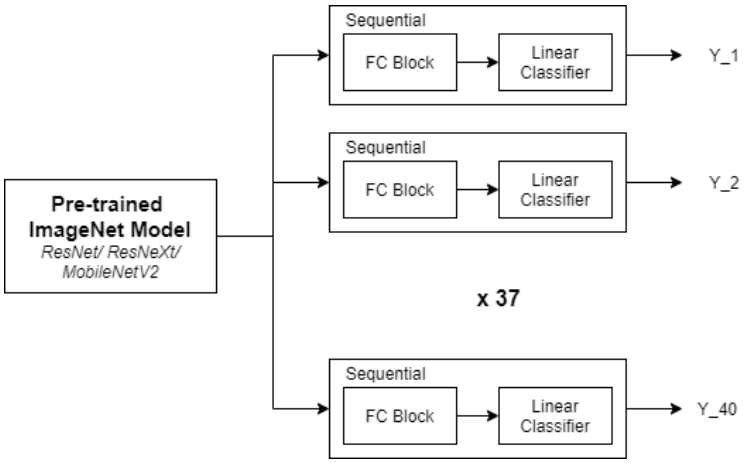


Figure 3: Network Architecture using ImageNet Pre-trained Models with multi-head Classifiers.

Three different model architectures were trialed for this assignment namely ResNet [3], MobileNetV2 [10] and ResNeXt [13]. Each models are initialized with using their corresponding pre-trained ImageNet weights before appending 40 attribute classifiers to the previous last linear layer. A simplified model architecture is shown in Figure 3. It is found that the deeper the model (more layers), the better the model performs. For instance, ResNet50 models consistently outperformed ResNet18.

2.3 Loss Function

$$CE(p_t) = -\log(p_t) \tag{1}$$

$$FL(p_t) = -\alpha(1-p_t)^{\gamma}\log(p_t) \tag{2}$$

Due to the class imbalance problem of the dataset stated in Section 1, there are more easy positives like *No_Beard* than hard positives like *Gray_Hair*. As a result, the model might tend to be biased towards learning more representations of the data-dominated class. Hence, by using Focal Loss [6] (Equation 2) over Cross Entropy Loss (Equation 1), it can help the network to learn sparse hard examples better. The hyperparameter α controls the weights of positive and negative samples, while γ adjusts the rate at which easy examples are downweighted. After some empirical tuning, $\alpha = 0.25$ and $\gamma = 3$ were set.

$$LS\_CE(p_t) = (1 - \epsilon)CE(p_t) + \epsilon \sum \frac{CE(p_t)}{N} \tag{3}$$

As recommended by He et al. [4], label smoothing, a regularization technique by Szegedy et al. [11], which perturbs the target variables, to make the model less confident of its predictions. It does so by performing a weighted linear combination of predicted values by an ϵ value [9]. This loss criterion (Equation 3) was trialed, but did not perform better than the Focal Loss.

2.4 Optimizer and Scheduler

SGD optimizer with momentum is used for most of the experiments with L2 weight decay set to 1e-4. The main change implemented was also suggested by He et al. [4] whereby the bias decay is explicitly turned off to avoid overfitting. The model is trained with the ReduceLROnPlateau scheduler from PyTorch, which dynamically reduce learning rate when the val loss did not improve over a pre-defined patience of 5 epochs.

3 Results

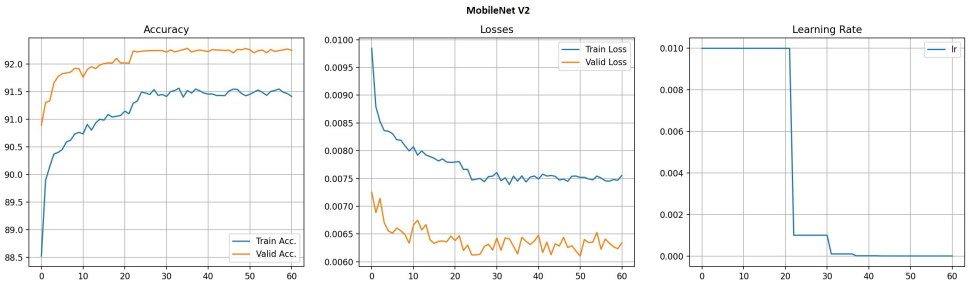


Figure 4: Accuracy, loss and learning rate curves of pre-trained MobileNetV2.

After performing data augmentations like MixUp, the training set will contain much harder examples than the validation set. This is reflected in Figure 4, whereby the Train Accuracy is consistently lower than Val Accuracy. This model attained a train accuracy of 91.55%, validation accuracy of 92.28% and test accuracy of 91.71%. It is possible that the network is underfitting and a deeper network is required to further improve the accuracy. However, subsequent training on larger networks did not yield better validation results.

3.1 Accuracy

The best model trained using pre-trained MobileNetV2 has average accuracy of 91.55%, 92.28% and 91.71% for train, validation and test set respectively. The second best scoring model uses pre-trained ResNeXt50 model [13] and it has an average accuracy of 92.88%, 92.10% and 91.65% for train, validation and test set respectively. Table 1 and Table 2 show the test and validation accuracy of each attribute.

	5_o_Clock_Shadow	Arched_Eyebrows	Attractive	Bags_Under_Eyes	Bald	Bangs	Big_Lips	Big_Nose	Black_Hair	Blond_Hair	Blurry	Brown_Hair	Bushy_Eyebrows	Chubby	Double_Chin	Eyeglasses	Goatee	Gray_Hair	Heavy_Makeup	High_Cheekbones
MobileNetV2	94.93	84.27	83.37	85.74	99.04	96.34	71.51	84.79	90.36	96.17	96.52	89.75	92.99	95.93	96.53	99.68	97.53	98.25	92.03	88.08
ResNeXt50	94.90	84.36	82.84	85.49	99.09	96.09	72.09	84.70	90.62	96.06	96.37	89.40	92.94	95.89	96.59	99.70	97.66	98.31	91.99	88.00
	Male	Mouth_Slightly_Open	Mustache	Narrow_Eyes	No_Beard	Oval_Face	Pale_Skin	Pointy_Nose	Receding_Hairline	Rosy_Cheeks	Sideburns	Smiling	Straight_Hair	Wavy_Hair	Wearing_Earrings	Wearing_Hat	Wearing_Lipstick	Wearing_Necklace	Wearing_Necktie	Young
MobileNetV2	98.61	94.09	97.05	87.84	96.49	76.84	97.27	77.83	94.05	95.31	97.97	93.32	84.78	85.30	90.83	99.09	94.28	87.75	96.97	89.01
ResNeXt50	98.63	94.22	97.01	87.77	96.54	76.39	97.30	77.76	93.94	95.18	98.01	93.26	84.50	85.17	90.93	99.12	93.79	87.81	96.97	88.75

Table 1: Models' test scores accuracy for each attribute

	5_o_Clock_Shadow	Arched_Eyebrows	Attractive	Bags_Under_Eyes	Bald	Bangs	Big_Lips	Big_Nose	Black_Hair	Blond_Hair	Blurry	Brown_Hair	Bushy_Eyebrows	Chubby	Double_Chin	Eyeglasses	Goatee	Gray_Hair	Heavy_Makeup	High_Cheekbones
MobileNetV2	94.69	86.67	82.02	85.25	99.12	96.15	85.28	83.64	92.12	96.04	96.82	86.34	93.07	95.88	96.97	99.6	96.94	98.17	92.99	88.99
ResNeXt50	94.45	86.65	81.67	84.70	98.96	96.16	83.51	83.76	91.70	95.81	96.84	85.47	92.96	95.69	96.64	99.6	96.90	98.04	92.93	88.93
	Male	Mouth_Slightly_Open	Mustache	Narrow_Eyes	No_Beard	Oval_Face	Pale_Skin	Pointy_Nose	Receding_Hairline	Rosy_Cheeks	Sideburns	Smiling	Straight_Hair	Wavy_Hair	Wearing_Earrings	Wearing_Hat	Wearing_Lipstick	Wearing_Necklace	Wearing_Necktie	Young
MobileNetV2	98.90	94.28	96.45	93.92	96.53	76.61	97.01	77.99	94.90	95.23	97.55	93.56	85.55	87.07	92.19	99.11	93.15	89.21	96.62	88.68
ResNeXt50	98.95	94.45	96.22	93.50	96.30	76.32	97.06	77.74	95.02	95.15	97.42	93.42	85.21	86.84	92.34	99.10	92.93	89.48	96.64	88.40

Table 2: Models' val scores accuracy for each attribute

4 Private Testset Analysis

4.1 Dataset Exploration

The private testset was provided to generate predictions using the best performing model for leaderboard scoring. Without the labels, a simple EDA was performed to understand the private testset. Note that this process was performed before the training of the final model. It was found that there are two duplicated images in this private testset under different Celebrity names. Figure 5 shows the estimated private testset distribution using the prediction labels from a preliminary model.

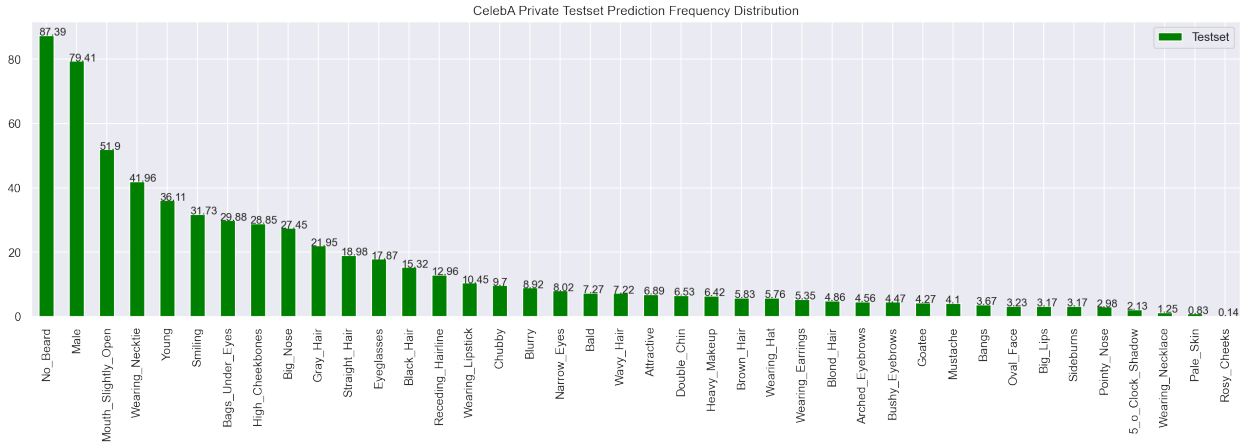
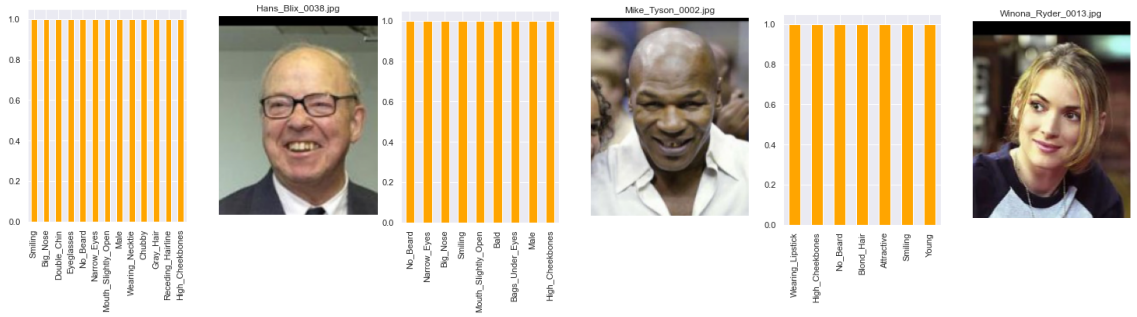


Figure 5: Frequency distribution plot of private testset using predictions from MobileNetV2.

As compared with Figure 1, the private testset has a different distribution from the public testset with increased percentages in *Male*, *Wearing_Necktie* and *Gray_Hair*. This is likely due to the dominating number of mature men images like *George_W_Bush*, *Colin_Powell* and *Tony_Blair*. It is apparent that this challenge places emphasis on rarer attributes in public set such as *Gray_Hair*.

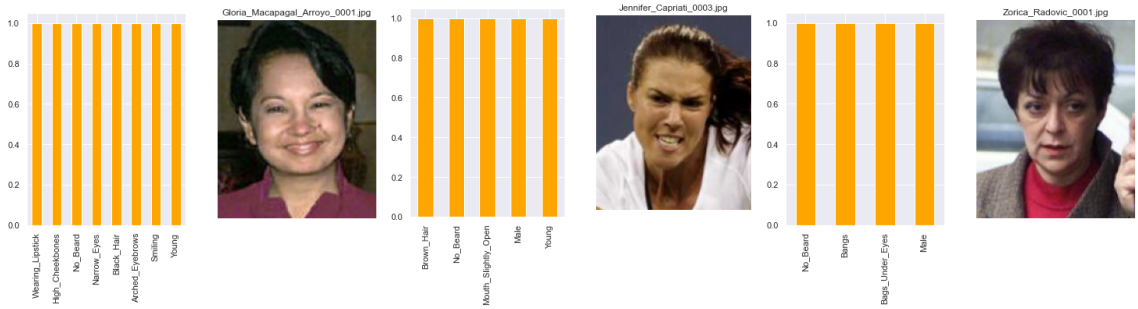
4.2 Prediction Analysis

As the private testset images are of size 250×250, they are first resized to 218×218 (long edge of Celeba public images) before center cropped to 198×158 like the training images. Figure 6 shows some positive examples of mostly correct labelled images, while Figure 7 revealed some hard-to-predict celebrity faces for the model from random sampling.



(a) Positive Example 1 (b) Positive Example 2 (c) Positive Example 3

Figure 6: Positive examples and their predicted attributes



(a) Mispredicted "Young" (b) Mispredicted "Male" (c) Mispredicted "Male"

Figure 7: Negative examples and their predicted attributes

5 Conclusion

This report documented the various data augmentations trialed, with random horizontal flip, random affine transformations (shift, scale and rotate) as well as MixUp training performing the best out of all experiments. Different deep convolutional pre-trained model architectures were also experimented using both Focal Loss [6] and Label smoothing Cross Entropy loss [11]. Out of the 50+ experiments, the best performing model architecture uses pre-trained MobileNetV2 [10] model with multi-head classifiers and was trained using Focal Loss. It attained accuracy of 92.28% and 91.71% on the Celeba's validation and test set respectively.

References

[1] Alexander Buslaev et al. "Albumentations: Fast and Flexible Image Augmentations". In: *Information* 11.2 (2020). ISSN: 2078-2489. DOI: 10.3390/info11020125. URL: https://www.mdpi.com/2078-2489/11/2/125.

[2] Manuel Günther, Andras Rozsa, and Terranee E Boult. "AFFACT: Alignment-free facial attribute classification technique". In: *2017 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE. 2017, pp. 90–99.

[3] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[4] Tong He et al. "Bag of tricks for image classification with convolutional neural networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 558–567.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[6] Tsung-Yi Lin et al. "Focal loss for dense object detection". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2980–2988.

[7] Ziwei Liu et al. "Deep Learning Face Attributes in the Wild". In: *Proceedings of International Conference on Computer Vision (ICCV)*. Dec. 2015.

[8] Hao Luo et al. "Bag of tricks and a strong baseline for deep person re-identification". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2019, pp. 0–0.

[9] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. "When does label smoothing help?" In: *Advances in Neural Information Processing Systems*. 2019, pp. 4694–4703.

[10] Mark Sandler et al. "Mobilenetv2: Inverted residuals and linear bottlenecks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 4510–4520.

[11] Christian Szegedy et al. "Rethinking the inception architecture for computer vision". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2818–2826.

[12] Sunil Thulasidasan et al. "On mixup training: Improved calibration and predictive uncertainty for deep neural networks". In: *Advances in Neural Information Processing Systems*. 2019, pp. 13888–13899.

[13] Saining Xie et al. "Aggregated residual transformations for deep neural networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1492–1500.

[14] Hongyi Zhang et al. "mixup: Beyond empirical risk minimization". In: *arXiv preprint arXiv:1710.09412* (2017).

[15] Zhun Zhong et al. "Random Erasing Data Augmentation." In: *AAAI*. 2020, pp. 13001–13008.