NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

CZ4042 Neural Network and Deep Learning Group Project

Project Idea: Gender Classification (D)

Group Members

Name	Matriculation Number
Dinh Phuc Hung	U1921644A
Oong Jie Xiang	U1920153J
Tran Hien Van	U1920891J

Table of Contents

Introduction

Automatic gender classification has become prevalent in many digital applications, whether in monitoring systems or on social platforms for image analysis. This demands the increase in performance accuracy of gender classification.

With the rise in computational power and availability of large datasets of labeled images, neural networks have been used to extract human attributes from images, such as gender and age. Recent advancements in image classification with the use of convolutional neural networks and new architectures may be useful in improving gender classification, and will be the focus in this project.

The goal of this project is two-fold:

1. Explore neural network architectures to improve the accuracy of gender classification
2. Consider age and gender recognition simultaneously to take advantage of the gender-specific age characteristics and age-specific gender characteristics inherent to images

The dataset used to benchmark models against is the Adience dataset, which is considered to capture the images as close to real world conditions as possible, including all variations in appearance, pose, lighting condition and image quality, to name a few. CelebA dataset, a large-scale face attributes dataset with more than 200K celebrity images, each with 40 attribute annotations and cover large pose variations and background clutter, is used only for pretraining.

Literature Review

Research is done on two subtopics: exploring different network architectures and hyperparameters to improve the accuracy of gender classification, and classifying gender and age with age-specific and gender-specific attributes.

Architectures Explored

Vision Transformer

Transformer is a state-of-the-art model that is primarily used for Natural Language Processing (NLP). It is designed to handle long-range dependencies while solving sequence-to-sequence problems. In Computer Vision (CV), Vision Transformer (ViT) [1] has been developed that produces excellent results while consuming far less computational resources than conventional convolutional neural networks.

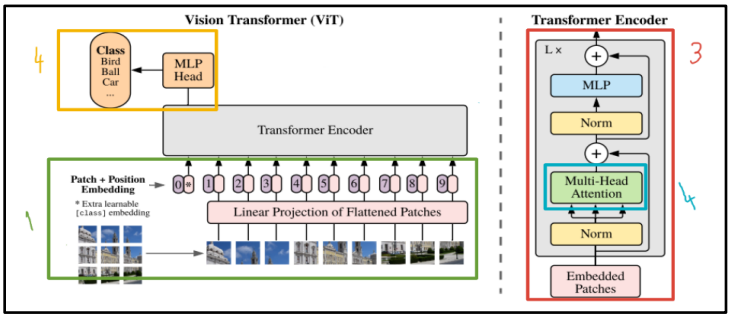


Figure 1: Vision Transformer architecture

ViT adopts a Transformer-like architecture and was initially trained for image classification tasks on ImageNet dataset. Like word embeddings in NLP, it represents an input image as a series of image patches including positional embedding. The resulting sequence of vectors is fed to a standard Transformer encoder to output the predicted class.

ViT can aggregate global features by adopting the self-attention mechanism of transformers. Contrary to CNNs, convolution is used to aggregate information in an inductive manner. In addition, Transformers are designed to handle various modalities (e.g., images, videos, text, and speech) using similar processing blocks. It also supports parallel computing to process input data, hence, the models outperform CNNs by almost four times when it comes to computational efficiency and accuracy.

In our project, we make use of transfer learning by fine tuning a Vision Transformer (ViT) model pre-trained on ImageNet-21k (14 million images, 21,843 classes) on our domain task. We chose the ViT-L32 variant - a large model that was trained on 32x32 image patches.

Residual Network

Residual network (ResNet) is developed by He et. al. for image recognition tasks. In image recognition, deeper layers are needed for the neural network to learn more complex representations from images. However, the addition of more hidden layers before the advent of this network leads to more training error due to the exploding and vanishing gradient problem, thus impeding the effectiveness of using deeper networks.

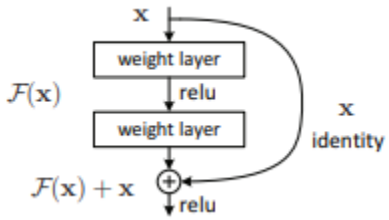


Figure 2: Building block for residual learning [2]

To tackle this problem, residual networks utilises identity mapping and skip connections. It adds the output from the previous layer to the next layer, so that even if the gradient obtained during backpropagation is extremely small, the neurons in that layer still learn the identity values. In this way, deeper networks can be created. Figure <> shows the building block for a skip connection. Even with 152 layers, ResNet still has less parameters and lower complexity than the VGG network, and attains lower error on ImageNet dataset, as the identity mapping does not incur additional parameters for the model.

In our project, we utilise transfer learning by fine tuning a ResNet model that is pre trained with ImageNet-21k on our domain task. We chose the ResNet-50 variant, a ResNet with 50 layers deep.

Inception Network

The Inception network leverages varying filter sizes, which are 1x1, 3x3, and 5x5, to increase the representational power. Additionally, **max pooling** is also performed. The outputs are **concatenated** and sent to the next inception module. The 1x1 conv filters learn cross channels patterns, which contributes to the overall feature extractions capabilities of the network.

Batch normalization is used extensively throughout the model and applied to activation inputs. Loss is computed using Softmax. Overall, Inception Module helps to extract features from input data at varying

scales through the utilization of varying convolutional filter sizes. In our project, we use Inception-v3 [3] with some improvements compared to version 1 to help decrease the computation cost. The architecture for Inception-v3 is shown below:

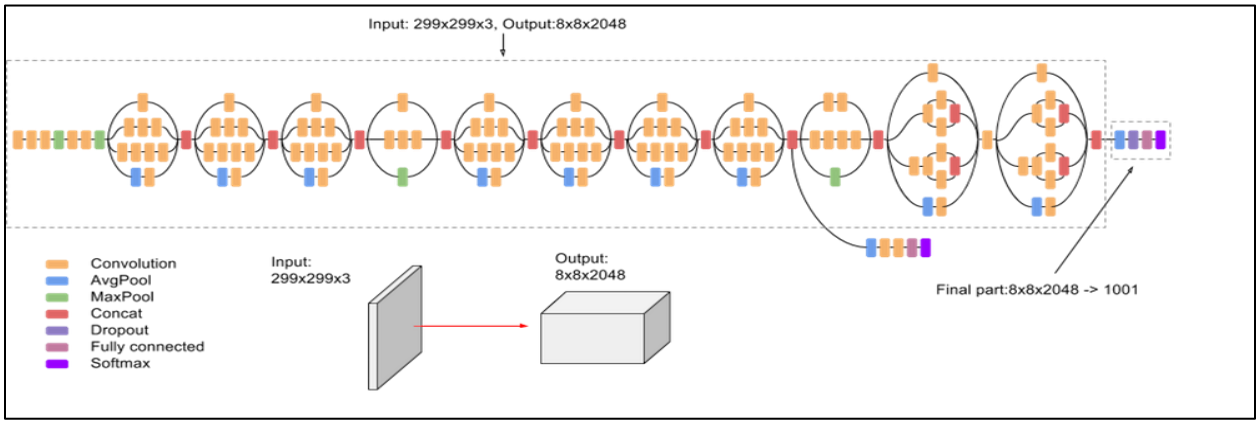


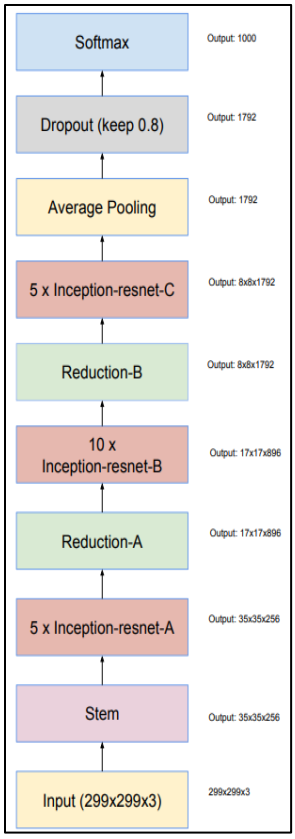
Figure 3: Complete Inception-v3 architecture

Residual-Inception Networks

Residual and inception networks presented above achieve good image recognition performance while using lower computational cost, compared to those without residual components and those with only a few stacked convolutional layers without dimension reduction. Residual-Inception network combines the Inception architecture with residual connections to accelerate the training of raw Inception networks and achieve better performance than its raw counterpart.

To deepen the Inception network, each Inception module is followed by 1 x 1 convolution to rescale the depth of the filters to match the depth of the input provided to the module. Skip connection is then added from the previous activation layer to the next activation layer, thereby preserving the gradients between Inception modules during backpropagation. For example, the Inception-resnet-A module below shows a direct connection from the previous Relu activation layer to the next.

Ideally, since the Inception-ResNet network combines the strengths of both Inception and ResNet, the performance is better than the individual networks. In this project, Inception-ResNet-v2 is used. This variant is a costlier hybrid Inception network with significantly improved recognition performance than Inception-ResNet-v1.



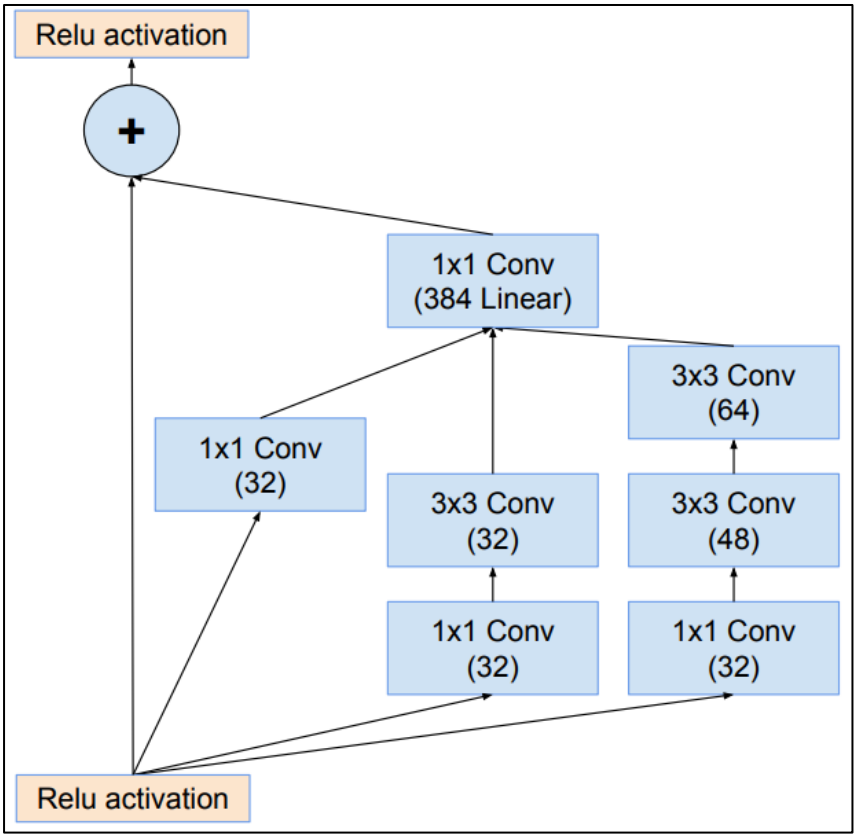


Figure 4 (left figure): Overall Schema for Inception-ResNet-v2
Figure 5 (right figure): A module of Inception-ResNet-v2 network [4]

Methodology

Image recognition requires a large amount of data and deep networks to learn complex patterns. However, exploding and vanishing gradients become a problem as the network becomes deep. State-of-the-art neural network architectures are therefore explored.

Transfer Learning

As the Adience dataset only has about 27,000 images in total, direct training on this dataset is insufficient for the model to learn insights, and may risk overfitting. To address the lack of dataset, we consider using transfer learning of models from CelebA dataset. To compare the improvements after using transfer learning, the following steps are taken:

1. Direct training of models on Adience dataset
2. Train models on CelebA dataset, and then transfer learn the models to Adience dataset
3. Compare the results of steps 1 and 2

In steps 1 and 2, four base models in the literature review are used in building the models: ResNet-50, InceptionV3, Residual-Inception-V2 and ViT-L32 variants.

In step 1, each base model is first loaded with weights trained on the ImageNet dataset. Next, each base model is topped with two fully connected layers with 1024 and 512 neurons, followed by a softmax layer. This process creates four models for direct training on Adience dataset. All four models are trainable to

learn from Adience dataset, except for Inception base model, which performs better when the first 52 layers are frozen by disabling their weight updates.

In step 2, all four models have the same layer architecture as their counterparts in step 1. However, these models are pre-trained on CelebA training dataset before being trained on Adience dataset. Step 3 then compares the performance of both direct and transfer learning based on the test accuracy.

Gender-age Classification

Multi-task Learning

A new approach involving multitask learning has recently been developed to facilitate learning common representations and keep track of correlations among tasks. We can view multi-task learning as a form of inductive transfer which is built upon an initial assumption about the model and data property (a.k.a. inductive bias). In our case, we assume that the model can perform better on the gender detection task after having learned age estimation task. It has been proved in previous experiments that age-gender characteristics can help to improve the performances on both tasks compared to single-task learning [5], [6].

There are two most used techniques for multi-task learning which are hard and soft parameter sharing of hidden layers. This project is built upon hard parameter sharing of hidden layers technique so that most of the hidden layers will be shared between all tasks, only the output layers are different. Hard parameter sharing can help to reduce the risk of overfitting. To examine the effect of multi-task learning, we will create two main components: shared layers and task-specific layers [12]. In our architecture, we will use Inceptionv3 as the core layer for shared components and then add other layers on top of it as follows:

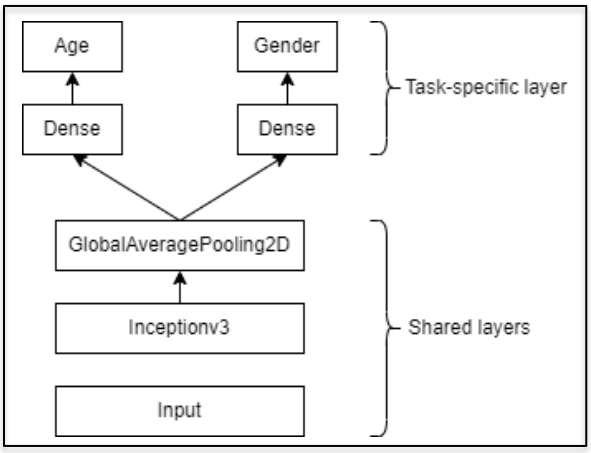


Figure 6: Multi-task architecture

Hyperparameter Tuning

At the beginning, we put inception model as the shared layer, and one fully connected layer for gender and age detection respectively. After which, we try out numerous methods to improve the detection accuracy as follows.

Default	What we do	Example	Result
1 shared layer of Inception-v3 no-top	Adding more shared layers	Add dropout layer	No improved
		Add batch normalization layer	Not improved
1 dense layer with 128 units	Adding more task-specific layers	Add 2 fully connected layers	Not improved
	Increase number of units	Change to 512 units	Not improved
Rmsprop optimizer	Changing the optimizer	Adam	Not improved
		SGD	Not improved
Set default learning rate (0.001)	Changing the learning rate	Add reduce learning rate callback	Improved gender accuracy

Finding Optimum Values with Hyperband

The table above illustrates that there are many configurations that affect the image recognition model's performance. While we have identified several of the key performance factors, it is nearly impossible to identify the set of hyperparameter values that creates the best model. Hyperparameter tuning is therefore used for age-gender classification.

Hyperparameter optimisation (HPO) can be classified into black-box optimisation and multi-fidelity optimisation [7]. Whereas in black-box optimisation where each possible model is run with the same resource, multi-fidelity optimisation attempts to reduce the evaluation cost by augmenting a few high-fidelity but expensive model samples with a large number of low-fidelity by resource-cheap model samples in a fixed resource budget (which could be time). Among the latter optimisation techniques, hyperband as a bandit-based approach helps in improving optimisation performance as it more intelligently allocates resources to more promising models, yet without giving the user the dilemma – spend more resources exploring more configurations which consequently yields less resources to tune each of them, or vice versa [8].

Hyperband uses Random Search in selecting the model configurations instead of intelligently converge to an optimal solution like Bayesian optimisation, which learns an expensive objective function and uses a probability model [9]. However, due to its ability to allocate resources to randomly sampled configurations, and the usage of early stopping when training each model, Hyperband can discover good sets of hyperparameters in a shorter time than Bayesian optimisation.

Data Augmentation

Common data preprocessing

We applied these two steps to all our models:

1. Normalize the pixel values to a [0,1] range.
   Having features on a similar scale could help gradient descent to converge faster towards the minima for Gradient Descent Based Algorithms such as neural network.
2. All images are resized to 192x192 pixels as almost all machine learning models expect a fixed-size training dataset of images.

Before feeding the image into the model, we also apply shear transformation and random zoom to increase the image data set.

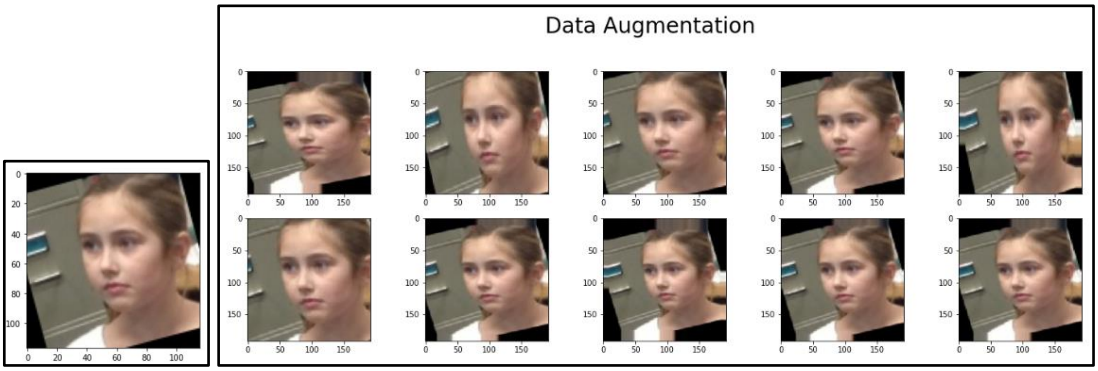


Figure 7: Data Augmentation example

RandAugment and MixUp techniques

Data augmentation has been shown to improve deep learning model generalization significantly. In this experiment, we want to explore the effects of augmentation pipeline on image classification. We adopt two commonly used techniques in CV: RandAugmentation and MixUp.

1. RandAugment [10]
   With a growing list of possible augmentations, people start randomly applying them (or subset of them), believing that a random set of augmentations would increase performances of models and replace the original set of images with a much larger number. RandAugment improves image classification results across a variety of datasets. In Keras-cv, this layer provides a standard set of augmentations on an image.
2. MixUp [11]
   MixUp is another augmentation technique that allows us to produce inter-class examples by interpolating the pixel values between two images. Models can be improved by avoiding overfitting the training distribution and generalizing to examples outside of the distribution. We ultilize the built-in functions from Keras-cv library.

These techniques are examined in a separated experiment. The result will be reported in later section.

Experiments and Results

Exploratory Analysis

The Adience dataset contains 26,580 photos across 2,284 subjects with a binary gender label and one label from eight different age groups, partitioned into five splits.

In our project, we did preprocess to remove “nan” values for age or gender, “unknown” gender and regroup the age group. After preprocessing, the total number of images is 17492 with the distribution for age and gender as follows:

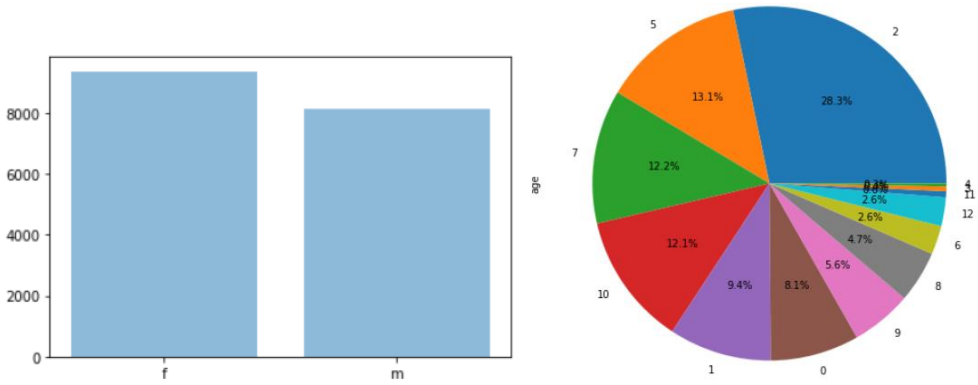


Figure 8: Data exploration results

Transfer Learning

	Architecture	Train	Validation	Test
Without transfer learning	Resnet	96.87%	89.14%	92.46%
	Vision Transformer	52.65%	55.52%	55.2%
	Inception-v3	96.91%	91.54%	92.46%
	Residual Inception	97.74%	94.45%	94.51%
With transfer learning	Resnet	96.30%	91.14%	91.49%
	Vision Transformer	76.5%	83.48%	82.34%
	Inception-v3	97.27%	93.83%	93.66%
	Residual Inception	94.75%	82.73%	95.09%

Gender and Age Classification

Age detection

	Architecture	Train	Validation	Test
1-task model	Inception-v3	94.87%	76.96%	73.37%
Multi-task model	Inception-v3	96.96%	70.95%	70.31%

Gender detection

	Architecture	Train	Validation	Test
1-task model	Resnet	96.30%	91.14%	91.49%
	Vision Transformer	76.5%	83.48%	82.34%
	Inception-v3	97.27%	93.83%	93.66%
	Residual Inception	94.75%	82.73%	95.09%
Multi-task model	Inception-v3	99.21%	95.78%	95.95%

RandAugment and MixUp techniques

We added RandAugment and MixUp into our data augmentation pipeline before training model. These figures show images before and after augmentation.

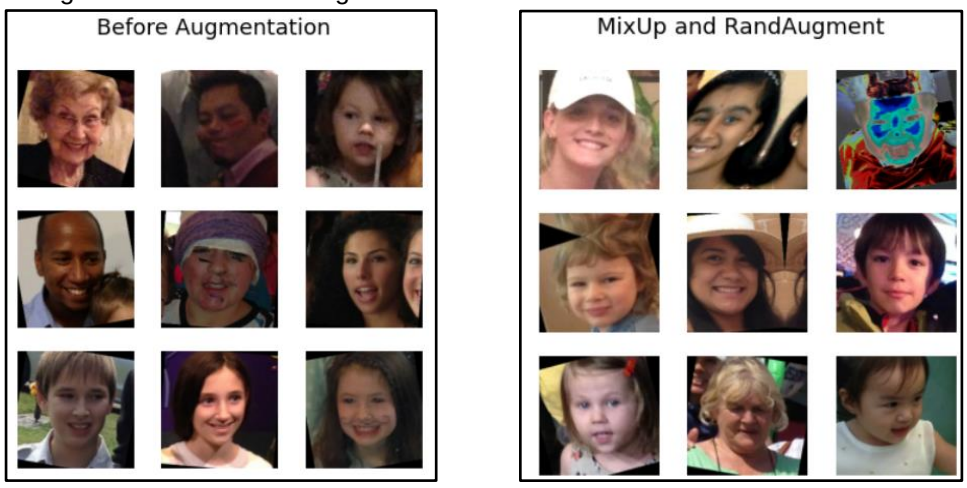


Figure 9: Data Augmentation example

In this experiment, we use Inceptionv3 as our main architecture to train on the newly augmented images. Below is the recorded results:

	Train	Validation	Test
Inceptionv3 pretrained with imagenet	47.21%	45.17%	46.8%

Discussion

Transfer Learning

Transfer learning helps leverage large models to save training time and improve the efficiency for training models. Since Inception-v3, Vision Transformers and ResNet are all trained with large dataset (ImageNet), their weights are useful for us to capture the most important aspects of the image so as to classify it accurately. As seen in the results, apart from

Gender-age Classification

Multi-task helps to increase gender detection but not age detection, which can be explained in future work. Adding more layers decreases the accuracy, showing that our models are overfitting. Surprisingly, adding regularizer and dropout layer does not help to reduce overfitting. Further work can be done to explain this observation.

RandAugment and MixUp techniques

The reported results reflect poor performances on the newly augmented training, validation and test set. Compared to the previous version of dataset without these augmentations, the accuracies are reduced almost half. Further work can be done to explain this observation.

Conclusion

In conclusion, transfer learning helps improve gender detection. Residual Inception v3 is the best architecture for training on gender prediction. Making use of age features help improve gender prediction but making use of gender features does not help improve age prediction.

For future work, we can make use of other features (race, emotion, etc) to detect gender. We can also apply age-gender classification for Resnet, Vision Transform and Residual Inception architecture to see if the age and gender accuracy will improve.

References

[1] A. Dosovitskiy *et al.*, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.” arXiv, Jun. 03, 2021. doi: 10.48550/arXiv.2010.11929.

[2] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition.” arXiv, Dec. 10, 2015. doi: 10.48550/arXiv.1512.03385.

[3] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision.” arXiv, Dec. 11, 2015. Accessed: Nov. 11, 2022. [Online]. Available: http://arxiv.org/abs/1512.00567

[4] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning.” arXiv, Aug. 23, 2016. doi: 10.48550/arXiv.1602.07261.

[5] D.-Q. Vu, T.-T.-T. Phung, C.-Y. Wang, and J.-C. Wang, “Age and Gender Recognition Using Multi-task CNN,” in *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Lanzhou, China, Nov. 2019, pp. 1937–1941. doi: 10.1109/APSIPAASC47483.2019.9023045.

[6] D. S. Al-Azzawi and D. S. Al-Azzawi, “Human Age and Gender Prediction Using Deep Multi-Task Convolutional Neural Network,” *J. Southwest Jiaotong Univ.*, vol. 54, no. 4, Art. no. 4, 2019, Accessed: Nov. 11, 2022. [Online]. Available: https://www.jsju.org/index.php/journal/article/view/323

[7] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, “Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization.” arXiv, Jun. 18, 2018. doi: 10.48550/arXiv.1603.06560.

[8] R. Elshawi, M. Maher, and S. Sakr, “Automated Machine Learning: State-of-The-Art and Open Challenges.” arXiv, Jun. 11, 2019. Accessed: Nov. 11, 2022. [Online]. Available: http://arxiv.org/abs/1906.02287

[9] M. J. Bahmani, “HyperBand and BOHB: Understanding State of the Art Hyperparameter Optimization Algorithms,” *neptune.ai*, Nov. 10, 2020. https://neptune.ai/blog/hyperband-and-bohb-understanding-state-of-the-art-hyperparameter-optimization-algorithms (accessed Nov. 11, 2022).

[10] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, “RandAugment: Practical automated data augmentation with a reduced search space.” arXiv, Nov. 13, 2019. doi: 10.48550/arXiv.1909.13719.

[11] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond Empirical Risk Minimization.” arXiv, Apr. 27, 2018. doi: 10.48550/arXiv.1710.09412.

[12] Jonathan Baxter, “A bayesian/information theoretic model of learning to learn via multiple task sampling,” Machine learning, vol. 28, no. 1, pp. 7–39, 1997.