# NANYANG TECHNOLOGICAL UNIVERSITY

# SCHOOL OF COMPUTER SCIENCE AND ENGINEERING


**Assignment for CZ4042**

**AY 2023-2024**

**Group members:**

| Name | Matric No. |
|---|---|
| Ong Kong Tat | U2022360K |
| Chong Yue Hong | U2120181D |
| Teo Han Hua | U2022463B |

# Table of Contents

# 1. Introduction

Gender classification has extensive applications in various fields, ranging from security and surveillance to targeted advertising and personalised user experiences. Traditional methods rely on handcrafted features and statistical approaches, while effective to a certain extent, often fail to capture the subtle and high-dimensional patterns in real-world data. The advent of neural networks, with their ability to learn hierarchical representations, has revolutionised this task, offering unparalleled accuracy through deep learning.

The primary objective of this project is to design and implement a robust neural network model capable of classifying gender from images with a high degree of accuracy. By utilising convolutional neural networks (CNNs) or transformers, the project aims to not only perform classification but also to explore certain advanced techniques to improve the performance of the model.

To train and evaluate the performance of our neural network, we employed a comprehensive dataset (Adience dataset) containing labelled instances with a balanced representation of genders. For image-based gender classification, the dataset will consist of facial images, potentially augmented with additional data to ensure diversity and robustness against overfitting.

We will iterate through 2 neural network architectures, implementing our own and another research paper's neural network, to understand their learning capabilities and limitations in the context of gender classification. The models will be trained, validated and tested using standard machine learning practices to ensure generalizability of the results. Special attention will be given to avoiding biases in the model to prevent skewing the classification performance in favour of one gender.

This project aims to contribute to the field of gender classification by developing a neural network that is not only accurate but also provides insights into the intricacies of gender representation in data.

# 2. Review of existing techniques

## 2.1 Introduction to Convolutional Neural Networks

Convolutional Neural Networks (CNNs) is a fundamental deep learning architecture that has emerged as a dominant force in the field of deep learning, revolutionising various applications in computer vision, natural language processing, and beyond. The review will be of both ResNet and alexnet, existing techniques of CNNs.

There are three main types of layers in CNN which are convolutional layer, pooling layer and fully-connected(FC) layer. The convolutional layer is the first layer of a convolutional network and can be followed by additional convolutional layers or pooling layers, the fully-connected layer is the final layer. The complexity increases with each layer, allowing it to identify greater portions of the image. It starts to detect the larger elements or shapes of the object until it is fully identified. The initial layers are mainly focused on colours and edges. Convolutional layer is where the majority of computation is and it is also the core building block of a CNN. Convolution involves a kernel or filter inside this layer moving across the receptive fields of the image, to detect the feature.

Over multiple iterations, the kernel sweeps over the entire image. After each iteration a dot product is calculated between the input pixels and the filter. The final output from the series of dots is known as a feature map or convolved feature. Ultimately, the image is converted into numerical values in this layer, which allows the CNN to interpret the image and extract relevant patterns from it [1].

Likewise for the pooling layer, it also sweeps a kernel or filter across the input image. However, the number of parameters in the input is reduced and it also results in some information loss. The point of the pooling layer is to reduce complexity and improve the efficiency of CNN [1].

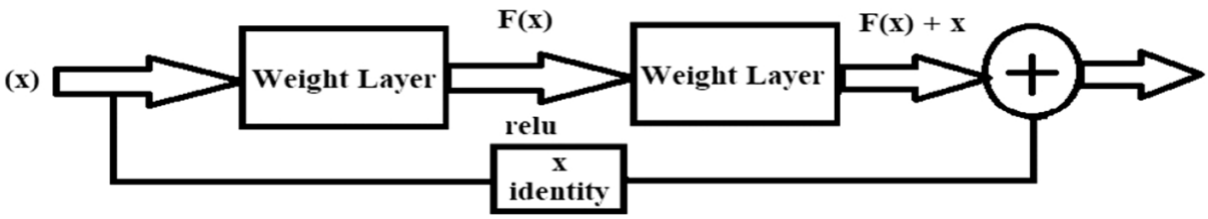
The last layer is the Fully Connected layer. Image classification happens in the fully connected layer based on the features extracted in the previous layers. All the inputs or nodes from this layer are connected to every activation unit or node of the next layer. Not all layers in the CNN are fully connected, as doing so would lead to an overly dense network. This approach would not only raise losses and degrade output quality but also incur substantial computational costs [1].

## 2.2 Resnet

Resnet which stands for residual neural network was proposed by K.M. He et al. in 2015 to solve the problem of vanishing/exploding gradients and decay in performance. The ResNet also performs decently in image classification tasks and has the capability of building extreme deep networks through residual units. ResNet is a traditional feed forward network with a residual connection. The core of the ResNet model is its residual unit, as shown below [2].

There are many variations of ResNetXX architectures where “XX” signifies the number of layers. Resnet18 is what we will use for this project. Resnet18 has around 11 million trainable parameters and it consists of convolutional layers having filters of size 3 × 3. Two pooling layers at the start and end are used to identify the connections between every two convolutional layers [3]. Shortcut connections skip some of the layers in the neural network and feed the output of one layer as input to the next layers [4]. It is also used by the ResNet18 to resolve the vanishing problem.

The two pooling layers are also not countable in the architecture of ResNet18. The figure below shows how the shortcut connections work in skipping layers.



# 3. Methodology

## 3.1 Preprocessing

- **Resize:** Resize the input images to 256 x 256
- **RandomCrop:** Crop the input images to 227 x 227 at a random location
- **RandomRotation:** Rotate the image by angle
- **RandomHorizontalFlip:** Horizontally flip the given image randomly with the probability of 0.5.
- **Normalise:** Normalise the data according to the mean and std of ImageNet dataset. mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]

When we have a complex model, the training might cause the model to be too specialised to the training data, resulting in low accuracy in predicting images outside from the training data. Data augmentation is an important technique to prevent model from overfitting because it will make the model focus on the features that are invariant for the classes [5]. We perform some data augmentation to the input images with the transformation provided above.

## 3.2 Network Architecture

We use the ResNet 50 model as the pretrained model for transfer learning [6].

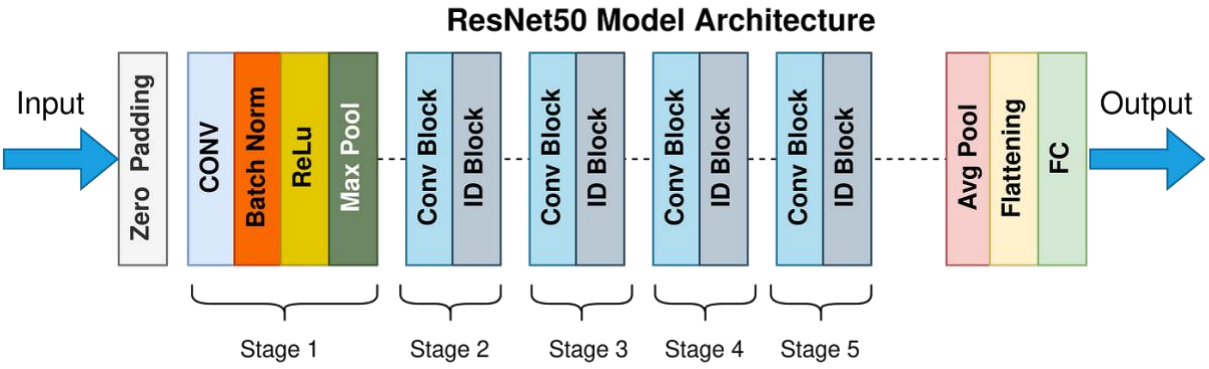



Figure 2: ResNet 50 Model Architecture

By applying transfer learning, the model does not have to be trained from scratch but already has some prior knowledge before the training with the new and unseen data [7]. We remove the last two layers (average pooling layer and fully connected layer) of the pretrained model, freeze the other layers, and add our custom Convolutional Neural Network on top of the pretrained model for transfer learning. The

newly added network refers to the network architecture created by Levi and Hassner [8]. The network architecture consists of three CNN layers and three fully connected layers. We reduce the parameters of the first fully connected layer by applying a larger size of the third max pooling layer. Below is the description of our network architecture.

1. Layer 1: 48 filters of size 2048 × 7 × 7 pixels. The output is then processed by a rectified linear operator (ReLU), a max pooling layer of kernel size 3 with stride equals to 2 and a local response normalisation layer.

2. Layer 2: The 96 × 28 × 28 output of the first layer is then fed to the second convolutional layer, containing 256 filters of size 96 × 5 × 5 pixels. Again, the output of the second convolutional layer is processed by a ReLU layer, a max pooling layer and a local response normalisation layer.

3. Layer 3: The 256 × 14 × 14 outputs by applying 384 filters of size 256 × 3 × 3 pixels, followed by a ReLU layer and an adaptive average pooling layer that will perform average pooling by automatically calculating the kernel size given the dimensions of the input.

4. Layer 4: After three convolutional layers, there are three fully connected layers. The first fully connected layer receives the output (384 x 1 x 1 pixels) of the third convolutional layer and contains 512 neurons, followed by a ReLU and a dropout layer.

5. Layer 5: A second fully connected layer that receives the 512 dimensional output of the first fully connected layer and again contains 512 neurons, followed by a ReLU and a dropout layer to prevent overfitting.

6. A third, fully connected layer which maps to the final classes for gender.

Finally, the output of the last fully connected layer is fed to a sigmoid layer which is used in binary classification where it will produce output between. The prediction itself is made by taking the class with highest value after the sigmoid layer for the given test image.

### 3.3 Pretraining

To have a better performance, we pretrain the model with images in CelebFaces Attributes Dataset (CelebA). This dataset consists of more than 200K celebrity images, with 40 attribute annotations [9]. The Align and Cropped Images (img_align_celeba.zip) in the dataset are used instead of the In-The-Wild Images to avoid the further alignment and preprocessing of the images. We only pretrain the model with 5000 CelebA images due to hardware and time restrictions. The data is split into a Training set and Validation set with the ratio of 7:3.

### 3.4 Actual Training and Testing

We split the data into 3 parts: Training set, Validation set and Testing set, with the ratio of 4:3:3 and set a batch size of 64 for the input images. The epoch is set to 8. Early Stopping mechanism is used during the training to prevent the overfitting of the training dataset. This mechanism will stop the training once the accuracy isn't improving on the validation dataset [10]. The patience value is set to 3, which means if the validation accuracy has not been increasing for three epochs, the training will stop.

# 4. Experiments and Results

## 4.1 Experiment 1

We train the pretrained Resnet50 model by freezing all layers and adding a sequential container. The sequential container consists of two fully connected layers with a RELU layer and dropout layer in between. By inputting the images in batches, the running times can be improved [11]. We test the accuracy of our model with the Adience dataset provided by The Open University of Israel [12]. This dataset has images taken without careful preparation and posting. The images are from Flickr albums, which are collected and uploaded by smartphone devices like iPhone 5 after the user opts in. There are some steps for image collection. They processed the photos downloaded from Flickr by running the Viola and Jones face detector and detecting the facial feature points [11].

The whole dataset collection comprises 26K images with 2284 subjects. Since our model focuses on gender classification, the number of images is 9848. Table 1 shows the breakdown of the images into different gender categories. We used the preprocessed dataset found on Kaggle [12].

| Set Name | Size of Set |
|---|---|
| Male | 4790 |
| Female | 5058 |

Table 1: The AdienceFaces Benchmark. Breakdown into Gender classes

The validation accuracy is shown in Table 2. It is shown that the accuracy increases during the training. The testing accuracy is **71.80%**

| Epoch | Validation accuracy |
|---|---|
| 1 | 55.28% |
| 2 | 70.58% |
| 3 | 68.18% |
| 4 | 70.58% |
| 5 | 71.09% |

Table 2: The validation accuracy of pretrained model training

### 4.2 Experiment 2

For the second experiment, which is also our final model, we train the combined model mentioned in Section 3. We pretrain the combined model with the CelebA dataset. After the pretraining, we freeze all pretrained layers except the last fully connected layer to perform transfer learning with Adience dataset. The epoch is set to 3. The validation accuracy is shown in Table 3.

| Epoch | Validation accuracy |
|---|---|
| 1 | 86.80% |
| 2 | 83.40% |
| 3 | 85.80% |

Table 3: The validation accuracy of custom CNN model pretraining with CelebA dataset

The epoch is set to 8 during the actual training. The validation accuracy is shown in Table 4.

| Epoch | Validation accuracy |
|---|---|
| 1 | 54.37% |
| 2 | 54.13% |
| 3 | 47.56% |
| 4 | 47.33% |
| 5 | 47.26% |
| 6 | 49.19% |
| 7 | 50.64% |
| 8 | 50.64% |

Table 4: The validation accuracy of custom CNN model training with Adience dataset

### 4.2 Results

We perform testing on the combined model. Table 5 presents the results for the testing of the model. The test accuracy is very low as compared to experiment 1. Figure 3 provides an example of gender misclassification. In conclusion, adding a custom CNN network on top of a pretrained model does not improve the overall performance of the model. It is likely that there is a flaw in our implementation of the custom neural network from [8] which resulted in such a low performance neural network.

| Test Accuracy |
| --- |
| 51.42% |

Table 5: The test accuracy of Experiment 2

Figure 3: An example of gender misclassification

# 5. Discussion

## 5.1 Limitation of Hardware

Due to limitations of hardware, there was insufficient time to optimise the hyperparameters and data augmentation techniques. On average, even just ten epochs take up to five hours and it is due to the hardware as it is run on google colab which provides 0 computer units, 12.7 GB for ram and 107.7GB for disk. We also run it on our physical computers but then it also takes up to 3 hours even with the number of workers increased and a context manager to manage external resources. Future improvements would require higher specs for the computer or virtual environment so as to further optimise the efficiency of experimentation.

## 5.2 Data Augmentation

There are many different ways to do transformation and we have only really tried one method of transforming the image. A few more examples of data augmentation that we can possibly execute or tune are geometric transformations, flipping, colour space, cropping, rotation, translation, noise injection and colour space transformations [15]. Tuning these parameters might increase the accuracy.

## 5.3 Hyperparameter Tuning

Due to the long runtime needed for every run, just tuning one parameter is very inefficient and might not yield any improvements. Possible improvements might require us to tune multiple parameters and experiment with it. A method for this would be k-fold cross validation. The efficacy of various hyperparameter settings is compared through cross validation by training the model and testing it for every possible hyperparameter tuning. The results are then summarised over k iterations to provide a performance score. This helps to reduce overfitting and provides a better idea of the model's performance. This is done by dividing the data into a number of subsets or "folds" and the model is trained and tested a number of times with a separate fold used for validation every time [16].

# References

[1] R. Awati, "techtarget," 24 Apr 2023. [Online]. Available: https://www.techtarget.com/searchenterpriseai/definition/convolutional-neural-network. [Accessed 9 Nov 2023].

[2] Y. D. Minghui Guo, "Classification of Thyroid Ultrasound Standard Plane Images using ResNet-18 Networks," IEEE, 2019.

[3] Fan, R.; Bocus, M.J.; Zhu, d Y.: Road crack detection using deep convolutional neural network and adaptive road crack detection using deep convolutional neural network and adaptive thresholding. IEEE Intell. Veh. Symp. pp. 474–479 (2019)

[4] S. T, "What are Skip Connections in Deep Learning?," 14 Aug 2023. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/08/all-you-need-to-know-about-skip-connections/.

[5] R. Wu, S. Yan, Y. Shan, Q. Dang, and G. Sun, "Deep image: Scaling up image recognition," arXiv preprint arXiv:1501.02876, vol. 7, no. 8, pp. 4, 2015.

[6] P. Huilgol, "Top 4 pre-trained models for image classification with Python Code," Analytics Vidhya, Aug. 9, 2023, [Online]. Available: https://www.analyticsvidhya.com/blog/2020/08/top-4-pre-trained-models-for-image-classification-with-python-code/.

[7] A. Hosna, E. Merry, J. Gyalmo, Z. Alom, Z. Aung, and M. A. Azim, "Transfer learning: a friendly introduction," Journal of Big Data, vol. 9, no. 1, p. 102, 2022.

[8] G. Levi and T. Hassner, "Age and gender classification using convolutional neural networks," in Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2015, pp. 34-42.

[9] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep Learning Face Attributes in the Wild," in Proceedings of International Conference on Computer Vision (ICCV), 2015.

[10] U. Vijay, "Early stopping to avoid overfitting in neural network- Keras," Medium, Oct. 13, 2020, [Online]. Available: https://medium.com/zero-equals-false/early-stopping-to-avoid-overfitting-in-neural-network-keras-b68c96ed05d9#:~:text=Early%20stopping%20is%20a%20method,improving%20on%20the%20validation%20dataset.

[11] E. Eidinger, R. Enbar, and T. Hassner, "Age and gender estimation of unfiltered faces," IEEE Transactions on Information Forensics and Security, vol. 9, no. 12, pp. 2170-2179, Dec. 2014.

[12] Geekjr, "Adience Dataset Preprocessed, Version 1," Retrieved Oct. 30, 2023, from https://www.kaggle.com/datasets/arcarcarc/adience-dataset-preprocessed/data, 2020.

[13] T. M. K. Connor Shorten, "A survey on Image Data Augmentation for Deep Learning," Journal of Big Data, 2019.

[14] R. Hendricks, "deepchecks," [Online]. Available: https://deepchecks.com/question/can-cross-validation-be-used-for-hyperparameter-tuning/#:~:text=Cross%2Dvalidation%20is%20used%20when,provide%20an%20overall%20performance%20score.. [Accessed 9 Nov 2023].