NANYANG
TECHNOLOGICAL
UNIVERSITY
SINGAPORE

SC4001 Neural Network & Deep Learning

Gender Classification

Oo Yifei	U2120933E
Lim Kang Wei	U2120531B
Yeoh Ming Wei	U2123351B

Table of Contents

Introduction

In the realm of computer vision and artificial intelligence, the ability to accurately classify gender from facial images has significant applications in various domains. The potential implications extend to fields such as social sciences, marketing, and human-computer interaction. In this report, we aim to design and refine a Convolutional Neural Network (CNN) model tailored for the task of gender classification. CNNs, renowned for their ability to exploit spatial features, is a viable approach to extract facial features even when the datasets are smaller in size and the face in the images are unconstrained [3]. Throughout the course of this project, we examine and implement a series of enhancements to our CNN architecture to improve the training results. Furthermore, we investigate the effectiveness of integrating deformable convolutional layers on model accuracy and enhancing the overall performance. Additionally, we investigate the potential impact of age classification on gender prediction to determine whether simultaneous age prediction enhances gender classification accuracy.

A review of existing techniques

Since 1996, multilayer neural networks have been used in the domain of gender classification. Inspired by the human nervous system, it is able to learn from training data even when confronted with low-resolution facial images. Notably, the Multilayer Neural Network achieved an impressive classification accuracy of 93% [5]. Introduced in the early 21st century, the Support Vector Machine (SVM) has emerged as a powerful method for pattern classification and regression. SVM's fundamental concept revolves around finding the optimal hyperplane that maximises class separation while minimising classification errors. Notably, SVM's accuracy surpasses that of other classifiers, particularly when addressing low-resolution images [5]. The Discrete AdaBoost algorithm, pioneered by Freund and Schapire (1996), was another approach to gender classification. Through a process of iteratively selecting informative features and combining them into weak classifiers, a robust ensemble model is constructed. Unique applications of "haar-like features" alongside threshold, mean, and Look-Up Table (LUT) weak classifiers have been explored [5]. In the year 2010, the K-Nearest Neighbour (kNN) algorithm was deployed for gender classification. However, it exhibited relatively lower classification rates compared to other contemporary techniques [5]. In 2012, gender classification took on the challenge of utilising Weber's Local Texture Descriptor, achieving near perfect results, particularly on the FERET benchmark [5]. The year 2015 marked a transformative moment with the application of CNNs to gender classification. This approach culminated in an accuracy rate of approximately 86.8% [2] on the Adience dataset [2]. In 2018, Hosseini et al. introduced an approach incorporating Wide CNN and Gabor filters, significantly enhancing gender classification accuracy. Their model achieved an accuracy of 88.9% on the Adience dataset [7].

Since then, the pursuit of constructing reliable age and gender estimation systems using CNNs often grapples with the challenge of obtaining large, accurately labelled datasets. Unlike expansive datasets such as ImageNet, those in the domain of social image-based age and gender classification are comparatively small. In this report, we would like to extend the research on the CNNs model in gender classification by introducing deformable convolutional neural networks (DCNNs), which was first introduced by Dai et al. to enhance the traditional CNNs in learning and incorporating spatial transformations in a more flexible manner [4]. Here, we will be applying this method to the tasks of age and gender classification. To our knowledge, there is no previous work that utilises DCNNs in gender classification tasks. The proposed architecture is intentionally compact, composed of three convolutional layers and two fully-connected layers, mitigating overfitting risks while aptly handling the binary classification of gender and multi-class age categorization. In our initialization phase, we diverge from practices using pre-trained models; our network is trained from scratch, relying solely on the benchmark's dataset. The network processes all three colour channels, takes input image size of 227x227, and follows a structured layer progression with dropout and normalisation to promote generalisation.

Description of the methods used

This report aims to evaluate the effectiveness of DCNNs in the task of gender classification and to evaluate the effectiveness of incorporating age into the model to extract age-specific gender features.

The dataset used to compare the model’s performance in this report is the Adience dataset [2]. This dataset consists of face photos in order to facilitate the study of age and gender recognition. The data collected is intended to be as true as possible to the challenges of real-world imaging conditions. In particular, it attempts to capture all the variations in appearance, noise, pose, lighting and more, that can be expected of images taken without careful preparation or posing. The figure below shows some example face photos of the Adience Dataset.

Figure of example face photos of the Adience Dataset

For this experiment, the pytorch framework [1] was used for transforming that data, creating the model and building the data loaders. In the following sections, we will describe how we prepared the data and load the Adience Dataset into our model. We will also introduce the key components used throughout our experiment.

1. Data discretization
    a. Instead of predicting the person's exact age, we labelled ages into different age groups based on the description of the Adience dataset [2]. These age groups are as follows: 0-2, 4-6, 8-13, 15-20, 25-32, 38-43, 48-53, 60-. The age groups are label encoded from 0 to 7 before fitting into the model.
    b. This process converts continuous data attribute values into a finite set of intervals and associates some specific data value with each interval.
2. Data Preprocessing
    a. Cleaning: removes rows with missing gender or age values and excludes rows with gender labelled as 'u' (undefined). We also constructed a column for 'directory' paths of the images and selected the age and gender columns to put in the final data frame.
    b. We define a data transformation pipeline using 'transforms.Compose', which includes 2 transformations: converting images to tensors and resizing them to a fixed size of 227 x

277 pixels, consistent with the dimensions employed in prior work [3]. During the conversion to tensors, the standardisation process uniformly scales the pixel values from the original range of [0, 255] to the normalised range of [0, 1].

   c. Next, we label encode the categories for gender and age.

3. Custom Dataset Class
   a. A class is defined to load and preprocess the dataset. The '__getitem__' method reads an image, applies the specified transformations, and returns the preprocessed image along with age and gender labels.
4. Data Loading
   a. An instance of 'CustomDataset' is created for the dataset using the defined transformation. This dataset is then used to create a DataLoader, which is where the batch size is determined to fit the model..
5. Optimizer and Loss Function
   a. The 'Adam' optimizer with initial learning rate set as 0.001 is used for training
   b. The loss function for gender is binary cross-entropy loss
   c. Since the age has been discretized into groups, the loss function used for age is cross-entropy loss.

To achieve our objective, the report consists of two main parts. The first part is to do cross-validation hyperparameter tuning to determine the optimal hyperparameter for the model. The second part is to construct different models with the optimal hyperparameters obtained and compare the results.

Experiments and Results

Hyperparameter Tuning: To build the optimal base model

Our project employed a cross-validation to find optimal hyperparameters for the CNN model. The dataset is loaded into 5 different folds, each containing a different subset of the data. A loop is created to iterate through the 5 different folds of data. For each fold, the model is trained for 5 epochs. The validation accuracy of the last epoch for each fold is recorded. This process results in five validation accuracy values, each corresponding to one of the five data folds. We computed the mean validation accuracy for each hyperparameter setting. Next, we identify the hyperparameter setting with the highest mean validation accuracy and set it as the optimal hyperparameter for our model. The hyperparameters chosen to be tuned are batch sizes, number of kernels of the CNN, and the size of the kernels.

- **Batch size**
  - The batch size of 16, 32, 64 and 128 are set for cross validation hyperparameter tuning.
  - Performance: Batch size 64 achieves the highest mean accuracy (0.7800). It has the best classification performance among the batch sizes tested.
  - Efficiency: While larger batch sizes (128) might converge quickly due to fewer weight updates per epoch, they don't perform as well in terms of accuracy. Smaller batch sizes (16, 32) in general might provide slightly better accuracy, but they might converge more slowly due to more frequent weight updates. Batch size 64 strikes a good balance between training efficiency and performance in our case.
  - Generalisation: A batch size that is too small (e.g., 16) may lead to noisy updates and overfitting. A batch size that is too large (e.g., 128) may lead to a loss of generalisation. Batch size 64 is often found to be a good compromise for many deep learning tasks in terms of generalisation.

- Stability: A batch size of 64 provides a relatively stable training process. It's less likely to get stuck in local minima or experience high variance compared to smaller batch sizes.
- Results: The figure below shows the mean validation accuracy across all 5 folds for different batch sizes.

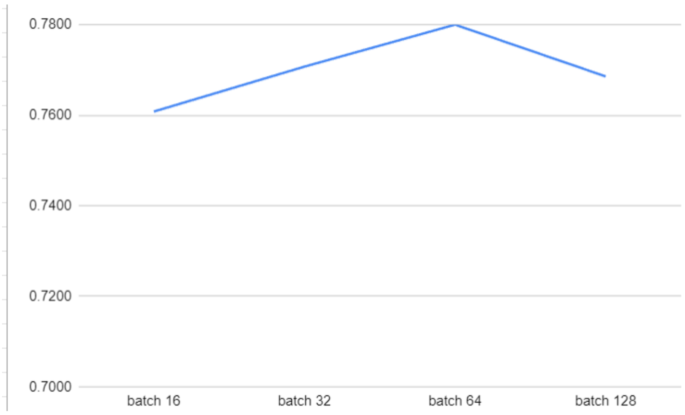

Figure of Mean Validation Accuracy for Different Batch Sizes

- **Number of kernel**
    - The number of kernels of 32 and 64 are set at the first convolutional layer for cross validation hyperparameter tuning.
    - Performance: The number of kernels set to 64 achieves a higher mean accuracy (0.7704) compared to 32 kernels (0.7682). It results in better classification performance.
    - Model Complexity: Increasing the number of kernels in a CNN allows the model to learn more complex features from the data. In this case, a model with 64 kernels captures more diverse and detailed features compared to a model with 32 kernels, which can lead to better performance.
    - Generalisation: While a model with more kernels may have a higher risk of overfitting if not properly regularised, in this case, it seems to generalise well and achieve a higher accuracy.
    - Computational Efficiency: The increase from 32 to 64 kernels does not significantly impact computational efficiency or training time, making it a practical choice.
    - Results: The figure below shows the mean validation accuracy across all 5 folds for different numbers of kernels at the first convolutional layer.

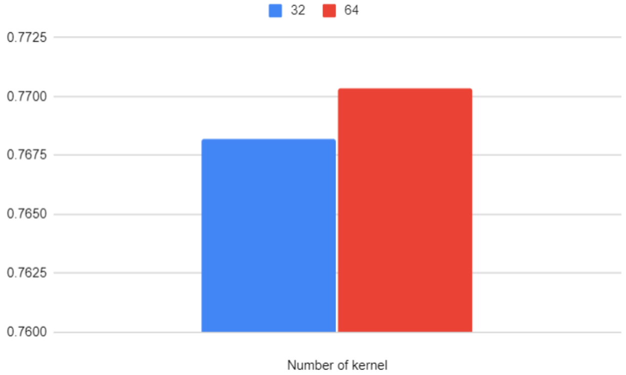

Figure of Mean Validation Accuracy for Different Number of Kernels

- **Kernel size**
    - The kernel size of 3, 5 and 7 are set for cross validation hyperparameter tuning.
    - Performance: Kernel size 5 achieves the highest mean accuracy (0.7767). It provides the best classification performance among the tested kernel sizes.
    - Feature Receptiveness: Larger kernel sizes (e.g., 7) can capture more global features in the input image, but they may also lose some fine details. Smaller kernel sizes (e.g., 3) can capture fine details but may not capture larger patterns as effectively. A kernel size of 5 strikes a balance between these two aspects, capturing both local and global features effectively.
    - Generalisation: A kernel size of 5 seems to generalise well. It performs better than the other tested kernel sizes, indicating that it captures relevant features for the gender classification task.
    - Model Complexity: While larger kernel sizes can increase model complexity, they may also increase the risk of overfitting. Kernel size 5 provides a good compromise between complexity and generalisation.
    - Consistency: Kernel size 5 provides consistent results with high accuracy. It's a stable choice for the task.
    - Results: The figure below shows the mean validation accuracy across all 5 folds for different kernel sizes.

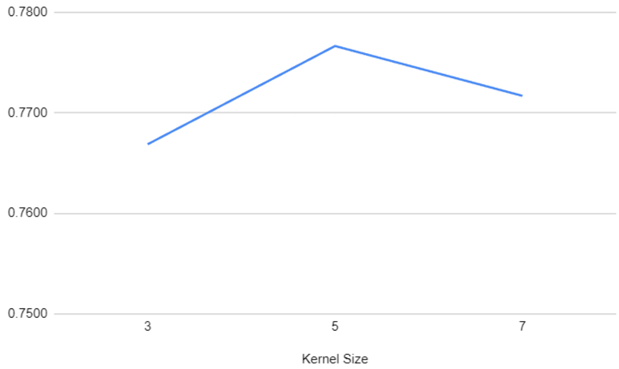

Figure of Mean Validation Accuracy for Different Kernel Sizes

Neural Networks Models

1. Model 1: Gender Only CNN Model

Model 1 is designed to classify gender based on face images and it serves as a base model for all the subsequent models discussed in this report. It's a straightforward CNN model that consists of three convolutional layers followed by two fully connected layers, and an output layer. In the first convolutional layer, 64 kernels are employed with a kernel size of 5, a stride of 1 and no padding. A subsequent 2x2 max-pooling operation is applied to the output of the convolution. The second and third convolutional layer share the same hyperparameter as the first, differing only in the number of kernels, which are set to 64 and 128 respectively. In the fully connected layers, the first layer consists of 256 neurons, while the second contains 128 neurons. Both of these layers have a dropout rate of 0.2, which means the model randomly zeroes some of the elements of the input tensor with probability 0.2 during training. The output layer utilises a sigmoid activation function with a single neuron, which predicts the gender as a binary classification task. Specifically, if the output value is larger than 0.5, then it classifies the face as male;

otherwise, it classifies the face as female. The binary cross entropy loss function is used here to train the weights with back propagation. The test accuracy of the final epoch of Model 1 is 0.77.

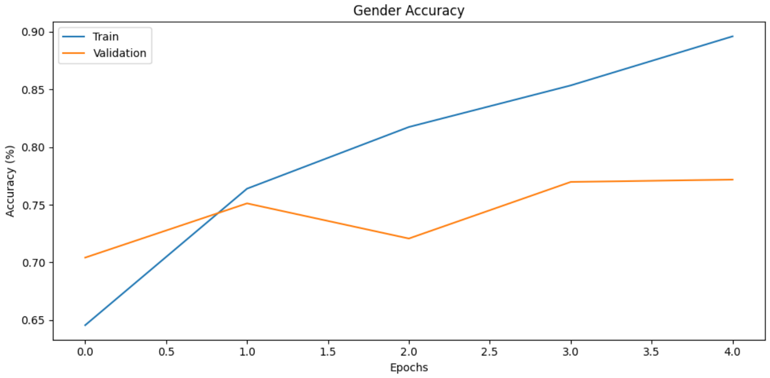

Figure of Gender Accuracy Against Epochs for Model 1

2. Model 2: Gender and Age CNN Model

This extensive model is designed for both gender and age classification. It's built on top of Model 1, but is able to predict both gender and age. It has 2 output layers, one for gender (binary classification) and another for age (multi-class classification). One of the output layers, tailored for gender classification as a binary task, shares the same structure as Model 1. The other output layer consists of 8 neurons corresponding to the 8 age groups and utilises a softmax activation function. The loss of the prediction is the sum of the binary cross entropy loss for gender and cross entropy loss for age. The loss is used to train the weights with the back propagation algorithm implemented by PyTorch. The features extracted by the convolutional layers are used to predict gender and age simultaneously in this model. The reason for this implementation is that we want to explore the convolutional layers' potential to extract age-specific gender characteristics and improve the overall performance of the model. The test accuracy of the final epoch of Model 2 is 0.76.

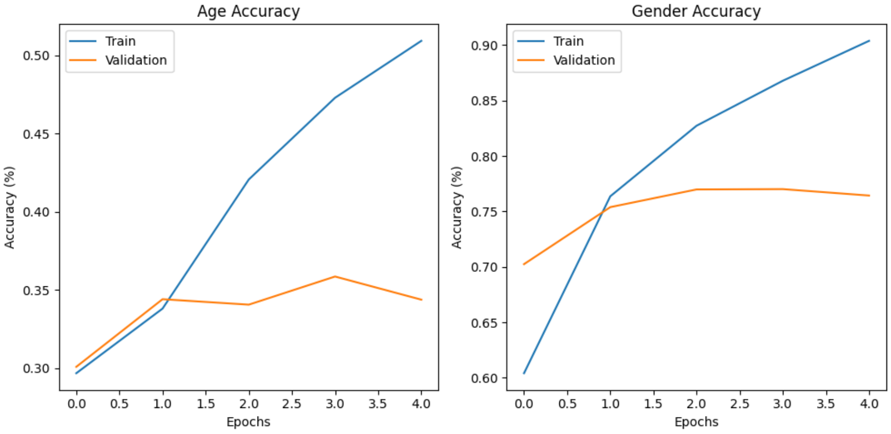

Figure of Age and Gender Accuracy Against Epochs for Model 2

3. Model 3: Gender and Age DCNN Model

Deformable Convolutional Neural Networks (DCNNs) take the robust spatial feature learning ability of standard CNNs and enhance it with deformable convolutional layers, allowing the network to adapt to geometric transformations. This makes DCNNs particularly advantageous at handling variations in object shapes and scale variations, making DCNNs having potential value for gender and age prediction tasks, where real world face images exhibit considerable size variations.

Model 3 shares a similar structure as model 2, with the distinction that all convolutional layers in Model 2 are replaced by deformable convolutional layers. During the forward pass, the DCNN processes the input through deformable layers to adaptively focus on salient features relevant to both gender and age predictions. This network outputs two predictions: one for gender and one for age. The loss function in Model 2 is likewise used here to train the parameters in the deformable convolutional layers and fully connected layers through back propagation. This combined loss function for gender and age ensures adaptive age-specific gender features are extracted by the deformable layers. The test accuracy of the final epoch of Model 3 is 0.79.

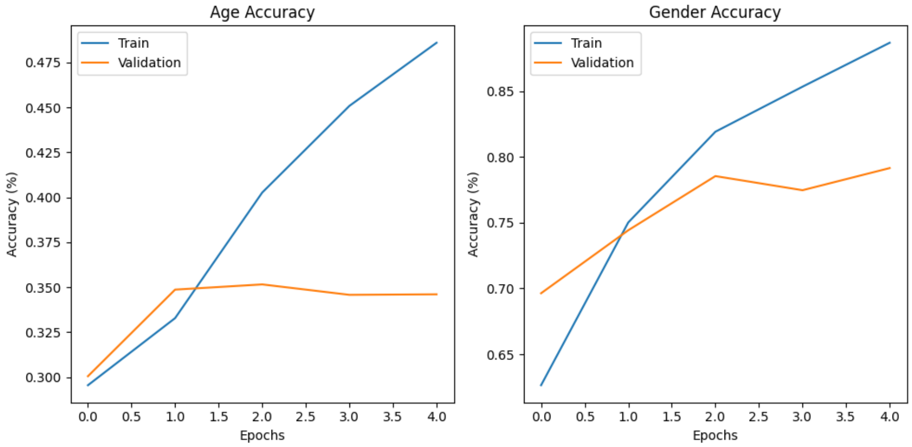

Figure of Age and Gender Accuracy Against Epochs for Model 3

To further assess the effectiveness of deformable convolutional neural networks in gender classification, we utilised data augmentation. This choice aligns with Dai et al.'s assertion that deformable convolutional layers address the limitations of conventional convolutional layers in modelling extensive and unknown data transformations [4]. This means that deformable convolutional layers may be able to extract more features without the need of data augmentation. Hence, we sought to investigate whether data augmentation alone can enhance the performance of DCNNs in gender classification tasks.

The data augmentation includes random cropping, horizontal flipping and random rotations up to 10 degrees. The objective is to employ data augmentation to artificially increase the diversity of the training dataset by applying a range of transformations to the original images, thereby simulating real-world variations. This approach may potentially enable the DCNN model to learn more generalisable and robust feature representations for gender classification tasks.

Apart from data augmentation, the training procedure is exactly similar to the process of training the DCNN model without data augmentation. The test accuracy of the final epoch under data augmentation is 0.74, significantly lower than the performance observed when data is not augmented.

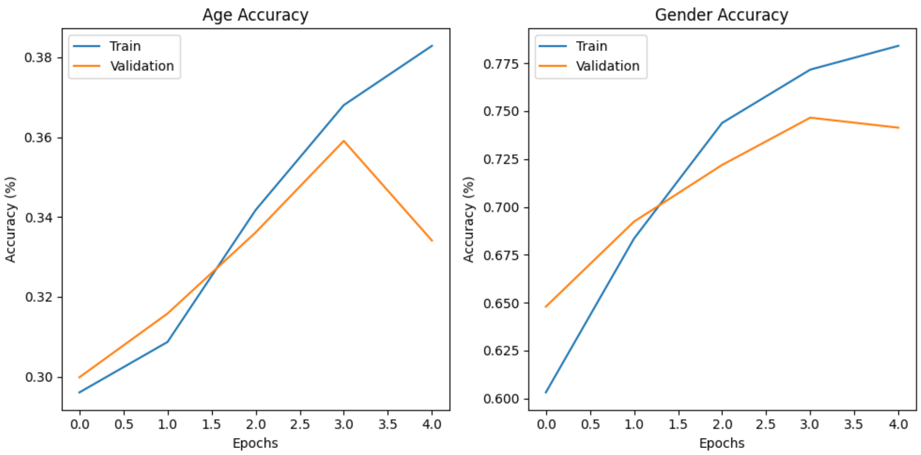

Figure of Age and Gender Accuracy Against Epochs for Model 3 with Data Augmentation

IMDB-WIKI Dataset

Another dataset that we used to compare the performance of the model is the IMDB-WIKI Dataset. It comprises a vast collection of 460,723 face images sourced from 20,284 celebrities from IMDb and 62,328 individuals from Wikipedia, thus 523,051 in total [5]. However, due to the dataset's large size, the creators mentioned that the assigned age information may be inaccurate. In this report, we randomly selected a subset of around 11,000 images for evaluating the model's performance. The primary distinction between the Adience Dataset and the IMDB-WIKI Dataset lies in the alignment and cropping of faces: Adience features well-aligned and cropped faces, while IMDB-WIKI presents faces with varying sizes. Figure below shows some example face images used in this report.

Figure of example face photos of the IMDB-WIKI Dataset

Data Preprocessing

The metadata for the face images is extracted from a .mat format file. A data frame that includes the image directories, gender and age information of the person inside was created with the extracted metadata. Preprocessing steps involved removing rows with negative age values or ages exceeding 100.

Additionally, rows with non-readable image directories were removed. Other preprocessing procedures are similar with those used for the Adience Dataset, as previously described.

Data discretization

Similar to the Adience dataset, the focus is on age group classification rather than predicting exact ages. The ages are categorised into different age groups as follows: 0-3, 4-8, 9-14, 15-20, 21-30, 31-36, 37-45, 46-53, 54-59, 60+. The age groups are label encoded from 0 to 9. This converts continuous data attribute values into a finite set of intervals and associates some specific data value with each interval to fit into the model.

Performance of Model 2 and Model 3

We fit the processed data into both Model 2 and Model 3 to evaluate the effectiveness of deformable convolutional layers in extracting features from faces of varying sizes. The test accuracy of the final epoch of Model 2 and Model 3 yielded results of 0.78 and 0.80 respectively.

Summary

The table above summarises the test accuracy for all the experiments.

Model and Description	Dataset	Test Accuracy
Model 1	Adience	0.77
Model 2	Adience	0.76
Model 3	Adience	0.79
Model 3 (with data augmentation)	Adience	0.74
Model 2	IMDB-WIKI	0.78
Model 3	IMDB-WIKI	0.80

Summary of Test Accuracies of Experiments

Discussion

Model 1 and Model 2 Comparison

The primary objective behind Model 2's implementation is to explore the convolutional layers' capacity to capture age-specific gender characteristics. This expansion aims to improve the model's overall performance by leveraging the information extracted from the convolutional layers. The test accuracy of the final epoch indicates a slight performance difference between the two models, with Model 1 achieving a test accuracy of 0.77 and Model 2 achieving a test accuracy of 0.76. The marginal difference in test accuracy suggests that the inclusion of age classification in Model 2 does not significantly impact gender classification performance. However, this outcome underscores the feasibility of using a single model for both tasks without a notable drop in accuracy. Furthermore, the results highlight the potential of convolutional layers to extract features relevant to both gender and age classification. The comparable performance of Model 2 to Model 1, despite the increased task complexity, emphasises the adaptability and generalisation capabilities of the convolutional layers. While the objective of improving overall model performance through the combination of gender and age classification within a single model is not entirely met, the results highlight the convolutional layers' capabilities in feature extraction for both gender and age classification tasks.

Model 2 and Model 3 Comparison

DCNN enhances traditional CNNs by introducing deformable convolutional layers. These layers allow the model to learn how to sample grid points during training, providing greater flexibility in capturing age-related features and gender characteristics. This addresses a limitation of regular CNNs, which are constrained by a fixed grid structure for feature sampling. In a standard CNN, a fixed rectangular window is used to sample from input feature maps at set locations, and pooling layers also use fixed sized windows to reduce spatial resolution. This approach has limitations, as it assumes that all features in a given CNN layer have the same size, regardless of the presence of objects at different scales in different positions. Deformable convolutions introduce a 2D offset to the regular grid sampling locations during convolution, allowing it to adapt and factor in the scale of different objects. This means that different parts of the feature map can have different receptive fields based on the scale of the object. When comparing Model 3, which incorporates deformable convolutional layers, to Model 2, which uses regular convolutional layers, the test accuracy of the final epoch shows a notable improvement. Model 3 achieves an accuracy of 0.79, while Model 2 achieves 0.76. This 3% increase in accuracy indicates that deformable convolutional layers have the potential to enhance gender classification performance by making the model more adaptable to geometric variations in face images.

Effect of Data Augmentation

Model 3, which incorporates deformable convolutional layers, achieves a test accuracy of 0.79 without data augmentation. However, when data augmentation is introduced, the test accuracy drops to 0.74. Data augmentation involves applying random cropping, horizontal flipping, and random rotations up to 10 degrees to artificially increase the diversity of the training dataset. The decrease in accuracy with data augmentation might be due to several factors. First, while data augmentation introduces diversity, it also adds noise to the training data, making the learning process more challenging. Second, the deformable convolutional layers might already be capturing some of the variations introduced by data augmentation, making additional augmentation less beneficial and may cause overfitting. These findings underscore the potential of DCNNs in gender and age classification and emphasise the importance of carefully designed data augmentation strategies to maximise model performance. Data augmentation is a double-edged sword. The effectiveness of data augmentation depends on the specific model and dataset, and finding the right balance is crucial for achieving optimal performance.

Performance of Model 2 and Model 3 on the IMDB-WIKI Dataset

The test accuracy on the IMDB-WIKI Dataset of Model 2 and Model 3 yielded results of 0.78 and 0.80 respectively. Remarkably, the outcomes of our evaluation on the IMDB-WIKI Dataset were in line with our initial expectations. As mentioned, the IMDB-WIKI Dataset, while vast and diverse, presented a challenge in the form of considerable variations in face sizes. This diversity can pose difficulties for conventional convolutional layers to effectively capture relevant features. On the other hand, the deformable convolutional layers allow the network to learn to focus on salient features that may not be uniformly aligned to a regular grid structure, which makes it able to adapt to the size variations in face images. The result that Model 3 performed better than Model 2 aligns with our initial hypothesis that DCNNs would excel in scenarios where data exhibits significant diversity, such as the varying face sizes encountered in the IMDB-WIKI Dataset.

Reference

[1] Paszke, Adam, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen et al. "Pytorch: An imperative style, high-performance deep learning library." Advances in neural information processing systems 32 (2019).

[2] Eran Eidinger, Roee Enbar, and Tal Hassner, “Age and Gender Estimation of Unfiltered Faces,” Transactions on Information Forensics and Security (IEEE-TIFS), special issue on Facial Biometrics in the Wild, Volume 9, Issue 12, pages 2170 - 2179, Dec. 2014

[3] Levi Gil, and Tal Hassner. "Age and gender classification using convolutional neural networks." In Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp. 34-42. 2015.

[4] Dai Jifeng, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. "Deformable convolutional networks." In Proceedings of the IEEE international conference on computer vision, pp. 764-773. 2017.

[5] Rai Preeti, and Pritee Khanna. "Gender classification techniques: A review." In Advances in Computer Science, Engineering & Applications: Proceedings of the Second International Conference on Computer Science, Engineering and Applications (ICCSEA 2012), May 25-27, 2012, New Delhi, India, Volume 1, pp. 51-59. Springer Berlin Heidelberg, 2012.

[6] Rasmus Rothe, Radu Timofte, and Luc Van Gool, “Deep expectation of real and apparent age from a single image without facial landmarks,” International Journal of Computer Vision, vol. 126, no. 2-4, pp. 144-157, 2018, Springer.

[7] Hosseini Sepidehsadat, Seok Hee Lee, Hyuk Jin Kwon, Hyung Il Koo, and Nam Ik Cho. "Age and gender classification using wide convolutional neural network and Gabor filter." In 2018 International Workshop on Advanced Image Technology (IWAIT), pp. 1-3. IEEE, 2018.