1 Data Selection

1.1 Data Description

1.1.1 Source of Data

The dataset is obtained from Kaggle, specifically from the 50k Celebrity Faces Image Dataset obtained from the link:
https://www.kaggle.com/datasets/farzadnekouei/50k-celebrity-faces-image-dataset. This curated subset consists of 50,000 celebrity face images, each in JPG format and sized 178x218 pixels. The original CelebA Dataset is well-known for its diversity, containing images with a wide range of poses, backgrounds, and facial attributes.

1.1.2 Dataset composition

For the purpose of this study, a subset of 650 images was selected from the original dataset. The selection criteria aimed to ensure diversity in terms of gender, age, and ethnicity. Specifically, an effort was made to include an approximately equal number of male and female images, with ages ranging from 18 to 40 years. Additionally, the subset aimed to cover various ethnicities represented in the original dataset.

1.1.3 Image Characteristics

Each image is in JPEG format and with image size: 178×218 pixels.

1.2 Labelling

Each of the 650 images was labeled with either 0 or 1, representing the observer's i.e. my subjective preference for the depicted face. The label 0 indicates a dislike for the face shown in the image, while the label 1 indicates a preference or liking for the face. The dataset includes labels that have been added to each images name, distinguished by the underscore _ character. For example, the original name of the first image is $0001.jpg$. After adding labels, the name is transformed to $0001_0.jpg$, where 0 indicates a dislike for the face shown in the image.

1.3 Data Visualization

The first three images together with their labels are shown in Figure 1:

2 Problem Statement

The goal of the project is to create a user-friendly face recognition model for dating apps. The project originated from a real-world scenario where a friend expressed frustration

Figure 1:

with spending too much time on dating apps, looking for attractive faces. The specific objectives and characteristics of the project are as follows:

2.1 User-Friendly Face Recognition Model

The primary aim is to develop a user-friendly face recognition model that can assist individuals in selecting potential matches on dating apps.

2.2 CNN Model using PyTorch

The chosen approach involves building a Convolutional Neural Network (CNN) using the PyTorch framework. This deep learning model is expected to learn from users' superficial preferences for faces.

2.3 Training the Model with User Preferences

The model is trained by collecting labeled data where users provide preferences (likes or dislikes) for multiple images. The labels are assigned based on the user's subjective evaluation of whether they like the face shown in each image.

2.4 Superficial Preferences Learning

The key assumption is that the model can learn the user's superficial preferences by analyzing the labeled images. The goal is to capture subtle features or patterns in facial appearances that align with the user's subjective notion of attractiveness.

2.5 Large-Scale Image Predictions

Once trained, the model is expected to make predictions on a large-scale dataset of images. This dataset represents a diverse set of faces, and the model's predictions are intended to assist the user in selecting faces they would likely find attractive.

2.6 Objective

Overall, the project focuses on leveraging deep learning techniques to automate the process of identifying attractive faces based on individual preferences. The user-friendly aspect implies that the model should be easy for users to interact with and provide meaningful recommendations for their dating app experience. The complexity of the task lies in training the model to understand and generalize from subjective preferences, as well as in the potential challenges of deploying such a model in a real-world dating app scenario.

3 Description and Justification of Methods and Analysis

The use of a Convolutional Neural Network (CNN) for the project on creating a user-friendly face recognition model for dating apps is justified for several reasons:

3.1 Image Recognition Task

CNNs are particularly well-suited for image-related tasks, and your project involves processing and analyzing images of faces. The hierarchical and spatial feature learning capabilities of CNNs make them effective in recognizing patterns and features in images.

3.2 Facial Feature Extraction

CNNs are designed to automatically learn hierarchical representations of features from data. In the context of face recognition, this is crucial as the model can learn to extract important facial features and patterns that contribute to attractiveness. CNNs excel at capturing intricate details in images.

3.3 Non-Linear Mapping

The problem of understanding subjective preferences for facial attractiveness is inherently complex and non-linear. CNNs, with their multiple layers and non-linear activation functions, are capable of capturing complex relationships within the data. This is essential for modeling the nuanced and subjective nature of human preferences.

3.4 End-to-End Learning

CNNs enable end-to-end learning, meaning the model can learn directly from raw input data (images) to predict output labels (likes or dislikes). This simplifies the overall pipeline and allows the model to learn relevant features directly related to the task at hand.

3.5 Model Flexibility

CNN architectures can be adapted and customized to suit the specifics of the problem. W can therefore experiment with different architectures, layer configurations, and optimization strategies to fine-tune the model's performance based on the characteristics of the given data set – In the course of this project, I have systematically experimented with various model architectures, conducting a thorough evaluation of their performances. Subsequently, I have judiciously selected the most suitable model architecture from the array of options under consideration.

3.6 Conclusion

In summary, the application of CNNs aligns with the nature of the problem – analyzing facial images for subjective preferences. The model's ability to learn hierarchical features, handle non-linear relationships, and its suitability for image recognition tasks make CNNs a strong candidate for addressing the complexities of understanding and predicting facial attractiveness in the context of dating app user preferences.

4 Interpretation and Reflection on Output

4.1 Model Description and comparison

Two CNN models (SimpleCNNModel and ComplexCNNModel) are built and compared in this project.

According to the python codes, the ComplexCNModel has a more complex architecture with additional convolutional layers, fully connected layers, and parameters. Notably, it lacks batch normalization and applies dropout only after the last fully connected layer. It uses ReLU activation for convolutional and fully connected layers, with Sigmoid activation for the last two layers. In contrast, SimpleCNNModel has a simpler structure. It includes batch normalization in both convolutional and fully connected layers and applies dropout after the first fully connected layer. Like the ComplexCNNModel, it uses ReLU activation for convolutional and fully connected layers, with Sigmoid activation for the output layer.

The choice between these models depends on factors like dataset characteristics and task objectives, as their effectiveness may vary. For this project, SimpleCNNModel outperformed due to several reasons. Firstly, it incorporates batch normalization, aiding in stabilizing and expediting the training process. Additionally, its simpler architecture can lead to better generalization, particularly when dealing with small datasets: in our case we only have 650 images. The strategic placement of dropout after the first fully connected layer in SimpleCNNModel is designed to prevent overfitting. Furthermore, SimpleCNNModel is less sensitive to batch size, making it more efficient with smaller datasets, while complex models might require larger batch sizes to generalize effectively.

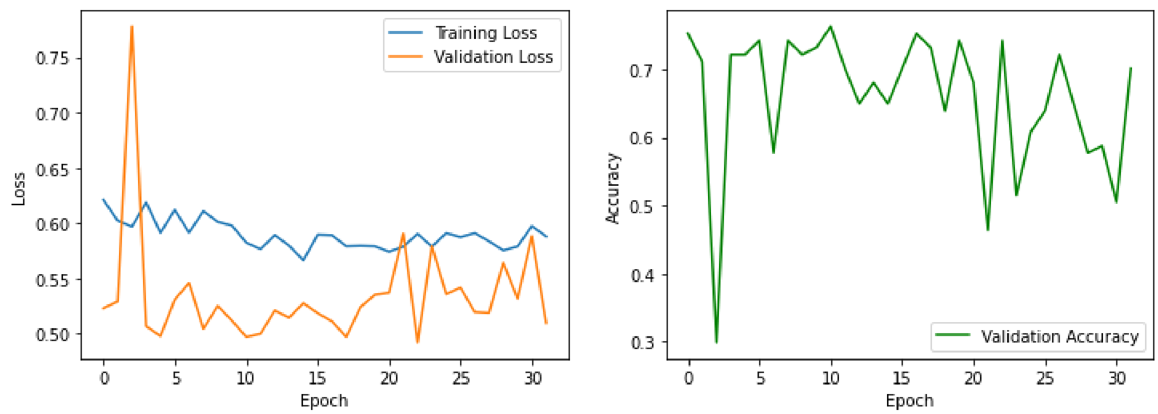

Figure 2:

4.2 Model Performance

The model's performance was evaluated over 32 epochs, and the training and validation results were as follows in Figure 2:

Furthermore, Figure 3 illustrated the model prediction result of a randomly selected 20 images from the test set. It can be seen from the image that: 15 out of 20 images are correctly predicted.

These results indicate that the model achieved a moderate level of accuracy on the validation and test sets. The fluctuation in validation accuracy throughout training suggests potential challenges in generalization, and further analysis or model adjustments may be needed for improved performance.

4.3 Data Characteristics

I resized images to 64x64 pixels and extracts labels indicating whether a face is liked (1) or disliked (0) from the filenames. The dataset is then divided into training, validation, and test sets. A special class is defined to handle the data, applying transformations like resizing and flipping to improve the model's learning.

The reason the simpler CNN model works better for our small dataset is because complex models, like the one we initially considered, can end up memorizing the small amount of data we have instead of learning general patterns. This memorization, known as overfitting, hurts the model's performance on new, unseen data. By choosing a simpler model, we help prevent overfitting and make the model more effective for our limited dataset.

4.4 Limitations and challenges

The limitations and challenges in our model performance stem from two main factors. First, the dataset itself is relatively small, comprising only 650 images. This small size

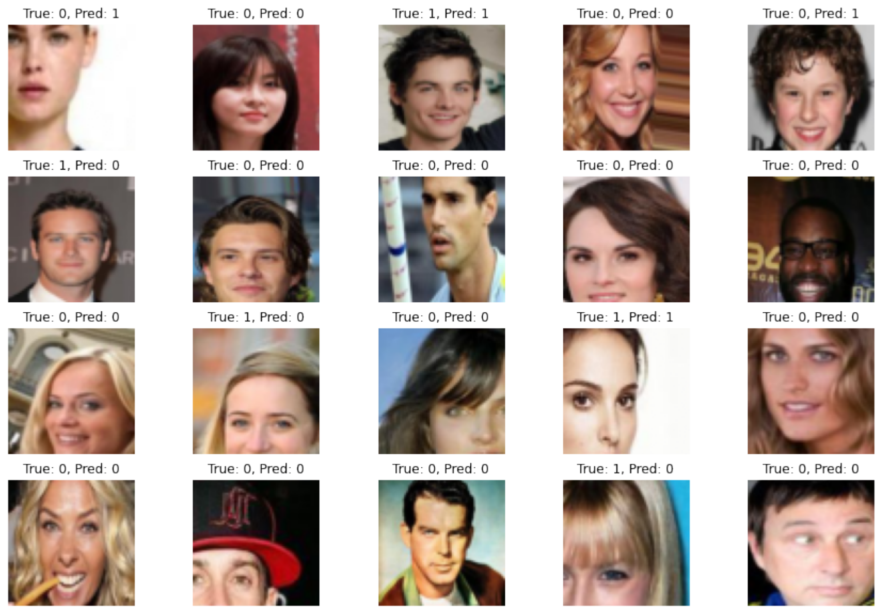

Figure 3:

poses difficulties for the model to learn robust patterns and generalize well to new data. Second, the imbalance between the two classes (likes and dislikes) further complicates the learning process. With only around 25% to 30% of the images labeled as liked, the model faces the challenge of dealing with unbalanced data. This imbalance affects the stability of the validation accuracy, as the model may become biased towards the majority class. Addressing these limitations, such as by acquiring a larger and more balanced dataset, could enhance the model's overall performance.

4.5 Potential improvements

Firstly, in order to address the limitations shown from the above section, we can try to acquire a larger and more balanced dataset, could enhance the model's overall performance.

Besides this, in order to make our model better, we can do a few things e.g. for the imbalanced data issue, we can use a weighted loss function that gives more importance to the minority class; experimenting with different hyperparameters, such as learning rate and dropout rate, may lead to better results; applying regularization techniques, like L1 or L2 regularization, can prevent overfitting. Indeed, I have tried all of these methods while none of them worked successfully in order to improve model performance.

For the next phase, I plan to explore ways to enhance our dataset's diversity. I'm considering employing advanced data augmentation techniques, such as rotation and color adjustments. Specifically, I'm interested in investigating whether implementing edge detection can be an effective approach to boost the model's performance.

4.6 Conclusion

In conclusion, our face recognition model for dating apps, built on a Convolutional Neural Network (CNN) using PyTorch, has shown promising results. However, it faces challenges due to the limited size of our dataset and the imbalance in the distribution of liked and disliked images. The model's performance is hindered by the difficulty of learning from a small and unbalanced dataset, leading to instability in validation accuracy. To address these limitations, future improvements could involve acquiring a larger and more balanced dataset and implementing advanced data augmentation techniques, such as edge detection, to enhance dataset diversity. Despite the challenges, this project lays the foundation for a user-friendly face recognition system that, with further refinements, could significantly benefit users in selecting preferred faces on dating apps.