Anna Gavrilova, Ruben Zink

Face attribute classification

Deep Learning

Anna Gavrilova, Ruben Zink

Introduction

Our project focuses on developing a Convolutional Neural Network (CNN) based deep learning model to accurately categorize various facial attributes, including age, gender, facial hair, eyeglasses, and headwear. Our model recognizes that facial data is complex, often showing multiple attributes in one image, and that is why it's designed to handle this complexity within a multi-label setup.

Previous solution

Previous solutions offers a spectrum of approaches, ranging from traditional computer vision techniques to more recent deep learning architectures. Recent advancements in deep learning have demonstrated the efficiency of convolutional neural networks (CNNs) for learning complex patterns directly from raw image data.

The prevoius solutions that have given inspiration to our model can be found bellow:

https://arxiv.org/pdf/1908.04913v1

https://arxiv.org/pdf/2403.12960v1

Dataset

The dataset we're using comes from the CelebA (Celebrities Attributes) database (https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html), created by a lab at The Chinese University of Hong Kong. It's a big collection of pictures of famous people, like actors and musicians, with labels describing different things about their faces.

In total, there are over 200,000 images in the CelebA dataset, each with labels telling us if the person in the picture is young or old, male or female, has facial hair, wears glasses, or has headwear on.

The dataset also includes some extra files, like "list_attr_celeba.txt", which give us more details about each image and its labels. These files help us organize and understand the data better.

To organize our data for training our model, we split it into three parts: training, validation, and testing.

- 80% of the data(162 079 images) is used for training,
- 10% (20 260 images) for validation to check how well the model is learning,
- the remaining 10%(20 260) for final testing to see how accurate our model is on new, unseen data.

Proposed method

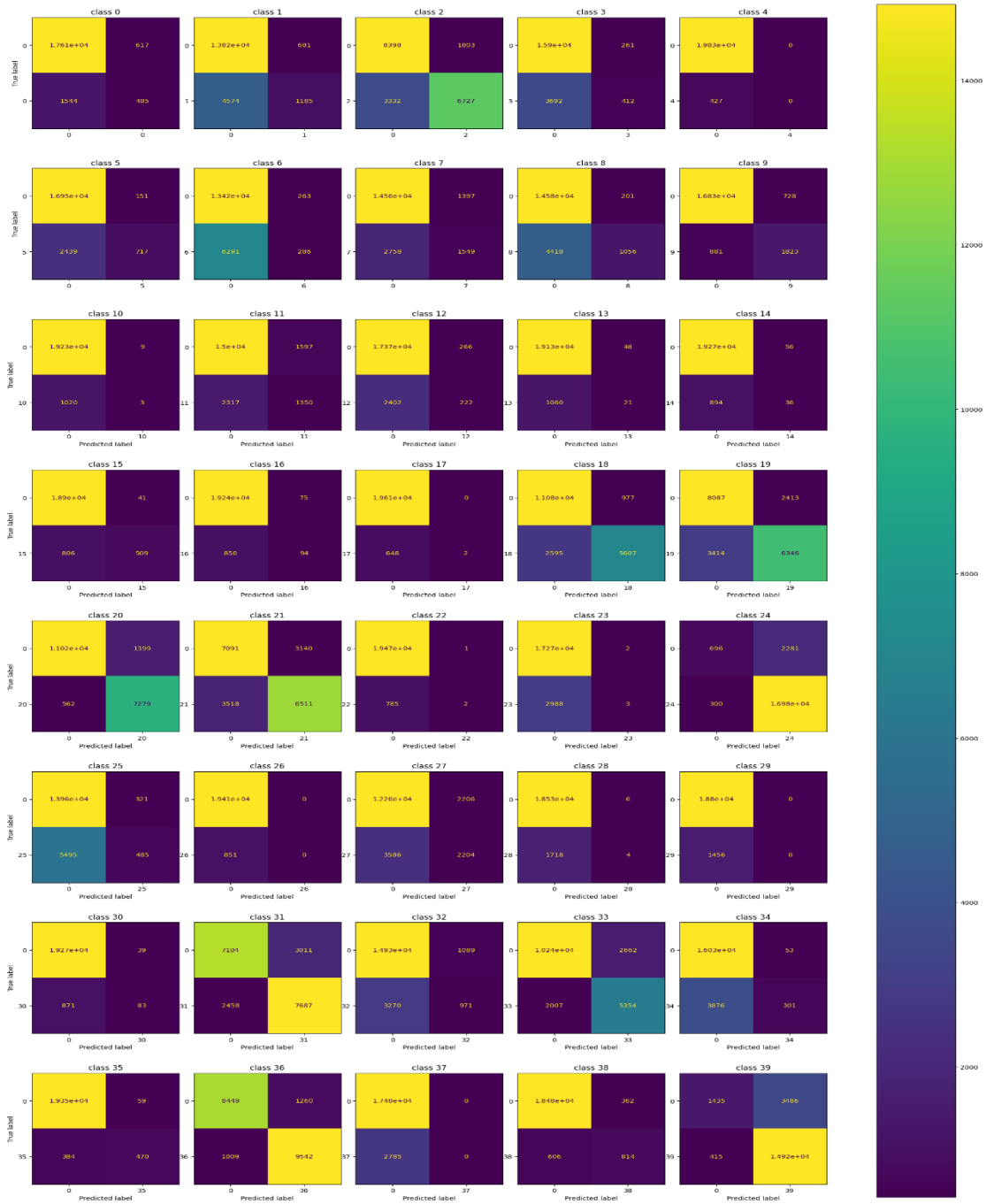
In order to complete the project successfully we tried 2 different methods. First of all, we created our own AI model with particular layers and logic. Afterwards, we decided to use the pretrained model (Inception V3) but with some adjustments for our task.

1. Our firtst method relies on using Convolutional Neural Networks (CNNs) to handle the complexity of facial attribute classification within a multi-label setup. CNNs are great at understanding images, making them perfect for analyzing facial data. With CNNs, our model was able to learn complex patterns in facial images, allowing it to classify various attributes. The task seemed to be quite complicated as it often shows multiple attributes in one image. To handle this difficulty, we were

to use proper layers with satisfactory activations (“relu”, “sigmoid”), optimizers (“adam”) etc. In order to improve the results, we also trained the model several times. We succeeded to use CNNs and multi-label classification to create a useable model for facial attribute classification.

2. The pretrained model we used with some adjustments, however, showed more accurate results.

For the purpose of showing the results, confusion matrixes were produced. At the end, we also created a simple website for using the AI model.



Evaluation method

In our project, we use a comprehensive evaluation method for multi-label classification tasks, where each sample may belong to multiple classes simultaneously. To begin with, we first separated different attributes and classes and evaluated the model's performance on each separately. To perform this part, we compared the number of right answers to the number of wrong. First, we set up lists to count how many times the model gets things right or wrong for each facial feature. Then, we go through the predictions our model made and compare them to the real labels. We count how many times the model got each feature right or wrong (the data is given in percentages).

Next, we calculate the accuracy for each feature. Accuracy tells us how often the model's predictions are correct out of all the predictions it made for that feature. To find accuracy, we wrote the code manually to better understand the metrics and its results regarding the model. We needed to divide the number of correct predictions by the total number of predictions made. This gives us a percentage for each feature, showing how accurate the model is for each one.

After that, we use a special method from a library called scikit-learn to calculate some overall scores, like the weighted F1-score, recall, and precision. These scores give us a big picture view of how well our model is doing overall. By following these steps and using these tools, we can really understand how our facial attribute classification model is performing and know where we can make it better.

Results and Discussion

All in all, we show that the model trained from our dataset produces balanced accuracy across some of the attributes. The reason of some errors and improper recognition may be caused by the dataset. The images often contained several attributes simultaneously, which hindered to perform correctly.

As a result, we are satisfied with our work. Not only have we made a useable, reliable AI model to recognize various facial attributes, but we also learnt much more about the CNNs, deep learning, different methods to solve one particular problem and evaluating them.