FYP Evaluation

Attribute Estimation Model

The models considered are Inferdo, DeepFace and BetaFace. Below are the results of these models in graphical form following execution on 50 randomly selected FairFace images. 50 was selected as the BetaFace API supports 50 images daily on the free track.

A graph with red and green squares

Description automatically generatedA graph with red and green bars

Description automatically generated

A graph of different colored bars

Description automatically generated with medium confidenceA white paper with black text

Description automatically generated

|  |  |
| --- | --- |
| Key | |
| Age | Ages detected correctly. |
| Race | Race detected correctly. |
| Emotion | Emotion detected correctly. |
| Gender | Gender detected correctly. |
| Detection | Images in which detection was carried out. |

A graph of a bar graph

Description automatically generated

Conclusion

In light of the fact that the Inferdo API does not possess race & emotion detection along with the fact that certain images could not be processed it was chosen to forgo using it.

Similarly, the BetaFace API was also not utilised as although it has the same capabilities as the DeepFace API it produces significantly worse in relation to race detection whilst only being marginally better at gender detection. Thus, it was determined appropriate that the DeepFace API would be utilised going forward.

DeepFace Alignment Backend

Throughout this section the FairFace dataset images previously utilised were reused. This was done as said images are accompanied with a ground truth which simplifies testing.

The available DeepFace face alignment backends are as follows:

* opencv
* ssd
* dlib
* mtcnn
* retinaface
* mediapipe
* yolov8
* yunet

However, excluding **opencv**, **mtcnn** & **retinaface**the remaining backends all proved problematic due, to raising errors throughout the implementation process. Thus, the testing carried out considers only the former three backends.

A graph of different colored bars

Description automatically generatedA graph of different colored bars

Description automatically generated with medium confidence

|  |  |
| --- | --- |
| Key | |
| Age | Ages detected correctly. |
| Race | Race detected correctly. |
| Gender | Emotion detected correctly. |

A graph of a bar chart

Description automatically generated with medium confidence

Conclusion

In conclusion from the minimal testing of the **opencv**, **mtcnn** & **retinaface** backends for the DeepFace.analyze() function the **mtcnn** was found to produce the best overall results. This can be seen from the tabulated data below:

**opencv**

* age 30% correct
* gender 56% correct
* race 68% correct

**mtcnn**

* age 28% correct
* gender 58% correct
* race 70% correct

**retinaface**

* age 24% correct
* gender 61% correct
* race 68% correct

This is also further supported by the following claim made in the [DeepFace](https://github.com/serengil/deepface) github repo:

* "RetinaFace and MTCNN seem to overperform in detection and alignment stages but they are much slower. If the speed of your pipeline is more important, then you should use opencv or ssd. On the other hand, if you consider the accuracy, then you should use retinaface or mtcnn."

The above claim is in line with the output provided in this notebook however retinaface seems to be producing subpar results given the particular FairFace dataset images.