gender Features research results

Data set:

|  |  |
| --- | --- |
| Size of image | classes |
| 7260 examples  + 19700 utk data set | glass  Mostatch  mask |

Methodology:

classifier: [InceptionResNetV2](https://keras.io/api/applications/inceptionresnetv2) encoder with there output branches

[glass, Mostatch, mask]

Data Split: 80% training, 20% validation

Loss Functions: focal loss

Metrics: Accuracy, [Recall, precision, f1 score] (macro and micro)

MODEL RESULTS

glass Class Report

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class | precision | recall | f1-score | support |
| No glass | 0.94 | 0.99 | 0.97 | 5483 |
| glass | 0.94 | 0.67 | 0.78 | 1053 |
| accuracy |  |  | 0.94 | 6536 |
| macro avg | 0.94 | 0.83 | 0.87 | 6536 |
| weighted avg | 0.94 | 0.94 | 0.94 | 6536 |

Mostatch Class Report

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class | precision | recall | f1-score | support |
| No Mostatch | 1.00 | 1.00 | 1.00 | 671 |
| Mostatch | 0.96 | 0.96 | 0.96 | 55 |
| accuracy |  |  | 0.99 | 726 |
| macro avg | 0.98 | 0.98 | 0.98 | 726 |
| weighted avg | 0.99 | 0.99 | 0.99 | 726 |

mask Class Report

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class | precision | recall | f1-score | support |
| No mask | 0.83 | 0.94 | 0.88 | 5507 |
| Hat | 0.99 | 0.95 | 0.97 | 1029 |
| accuracy |  |  | 0.99 | 6536 |
| macro avg | 0.94 | 0.83 | 0.87 | 6536 |
| weighted avg | 0.94 | 0.94 | 0.94 | 6536 |

## **The typical transfer-learning workflow**

This leads us to how a typical transfer learning workflow can be implemented in Keras:

1. Instantiate a base model and load pre-trained weights into it.
2. Freeze all layers in the base model by setting trainable = False.
3. Create a new model on top of the output of one (or several) layers from the base model.
4. Train your new model on your new dataset.

Note that an alternative, more lightweight workflow could also be:

1. Instantiate a base model and load pre-trained weights into it.
2. Run your new dataset through it and record the output of one (or several) layers from the base model. This is called **feature extraction**.
3. Use that output as input data for a new, smaller model.

## **Fine-tuning**

Once your model has converged on the new data, you can try to unfreeze all or part of the base model and retrain the whole model end-to-end with a very low learning rate.

This is an optional last step that can potentially give you incremental improvements. It could also potentially lead to quick overfitting -- keep that in mind.

It is critical to only do this step after the model with frozen layers has been trained to convergence. If you mix randomly-initialized trainable layers with trainable layers that hold pre-trained features, the randomly-initialized layers will cause very large gradient updates during training, which will destroy your pre-trained features.