Deep Learning Assignment 2021

Dr. Eva Vanmassenhove and Dr. Gorkem Saygili

Deadline: October 5 2021

1 Practical Aspects

The sections below described important practical aspects for the group assignment for the TiU Deep Learning Course, BLOCK 1, 2021.

1.1 Group Assignment

The assignment is to be made in (small) groups consisting of 3–4 people. Groups will **not** be pre-assigned by the lecturer. As such, you will be required to organise yourselves into groups. If you do not manage to join/form a group of **3–4** people within a week (i.e. by the **21st of September 2021**), you will be randomly assigned to an existing group of 3 (or merged with another group of less than 3 people). Groups formed with only 3 people should be aware of the fact that a 4th person might be added to their group.

1.2 Grading

The hands-on assignment will count for **30% of your total course grade**. The grades will be based on the quality of your work as judged by the instructor based on your report and code. Their might be hardware limitations in order to train very complex neural networks. You are encouraged to mention improvements to your classifier that you could implement if you had more resources available.

The assignment itself does not necessarily require access to high-end GPUs or a very powerful machine. If you do want to experiment with GPUs or TPUs, you could use the facilities provided by Google Colab[1].

Passing the assignment is not mandatory to pass the course but it is highly advisable. As it is not compulsory, **there will be no resit**. The exam may include questions that are easier to answer if you have worked actively on the assignment.

The assignment will be graded on 10 points, a detailed description of all the required components can be found in Section 2. The grades of each component are outlined below:

- A baseline model along with the required plots and performance metrics (1 point)

[1] https://colab.research.google.com

- An improved model, the required plots and performance metrics, an explanation and discussion of the experiments and the results obtained (6 points)
- The pre-trained model, the required plots and performance metrics, an explanation and discussion of the experiments and results obtained (2 points)
- Ethical concerns/considerations/limitations related to this particular task, i.e. binary gender classification based on images (1 point)

2 Task Description

The task of the assignment is known as (binary) gender recognition from face images. There are several papers focusing on this topic in the literature [1–4]. You will be using facial images from two Kaggle datasets as described in Section 2.1. After downloading and preparing the datasets, your assignment is to:

- Create a virtual environment and install tensorflow, SciPy, matplotlib, keras, seaborn libraries. These are the libraries we recommend but you can also install the ones of your choice.
- Implement the baseline CNN algorithm that is shown in Fig. 1. It is a network consisting of: a convolutional layer with 16 filters of size 3×3 with ReLU activations followed by a max pooling layer of size 2×2; another convolutional layer identical to the first followed by a max pooling layer; a fully connected layer of size 64 and ReLU activation function; and finally, an output layer of size 1 with a sigmoid activation function. The pooling layers have a stride of 1.
- Analyze the performance of the baseline by plotting: (i) the training and validation losses and accuracies on the training and test set, (ii) the Receiver Operator Characteristic (ROC) curve with the Area under the Curve (AUC) score and a confusion matrix for the validation and test set. Examples of accuracy and loss plots are shown in in Fig. 2, an example of a ROC curve and confusion matrix is shown in Fig. 3, respectively. Report performance measures (accuracy, sensitivity, specificity and F1-score).
- Once you have a baseline model, adapt/fine-tune the network to improve its performance by: (i) changing the hyper-parameters (e.g. add more layers) and/or (ii) applying data augmentation techniques. Illustrate the improvements of your new network over the baseline by: (a) plotting the ROC curve with AUC score and (b) reporting performance measures. Compare and explain the differences between the two models as well as potential reasons behind the increase in performance .
- Next, train a new model using transfer learning. Utilize VGG16 architecture for feature extraction. Freeze the layers until the fully connected layer such that these layers will not be updated through training. Add your fully connected layers (as many as you like) and present the results that you obtained on the test set (ROC curve with AUC

score, performance measures and confusion matrix). Comment on the performance with respect to the baseline and the network that you designed in the previous step.

Figure 1: Baseline CNN Algorithm

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 98, 98, 16)        448
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 49, 49, 16)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 47, 47, 16)        2320
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 23, 23, 16)        0
_________________________________________________________________
flatten (Flatten)            (None, 8464)              0
_________________________________________________________________
dense (Dense)                (None, 64)                541760
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 65
=================================================================
Total params: 544,593
Trainable params: 544,593
Non-trainable params: 0
```

2.1 Data set

The training and test data are available on Kaggle. The **training data** can be downloaded using the following link:

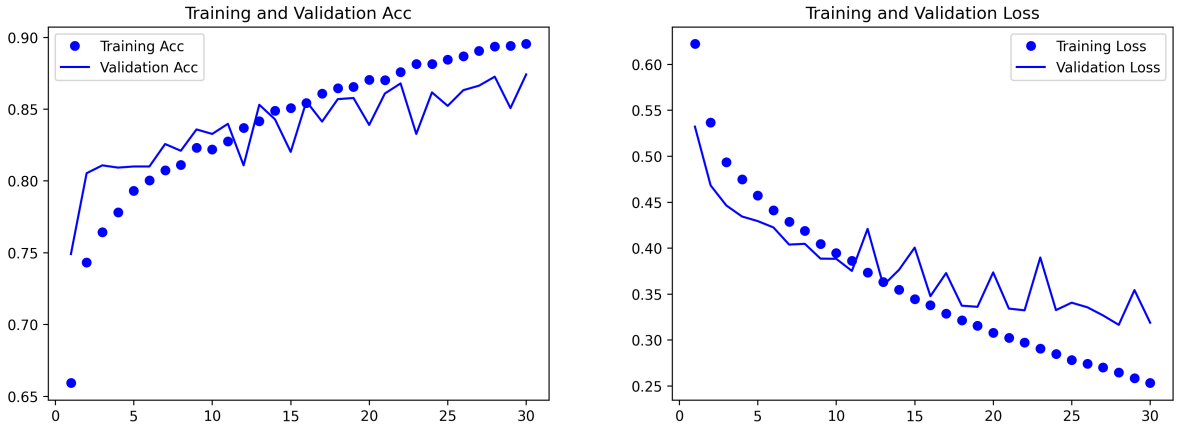
https://www.kaggle.com/rashikrahmanpritom/gender-recognition-dataset

The dataset is provided in train, validation and test folders. Although there is a separate validation folder, the files inside that folder do not have labels. Hence, you should use the images in the test folder as your validation set. The traning folder contains 11135 RGB images of size 100×100 and the test folder contains 1279 RGB images of the same size – you will use those for validation.
It is important to have a test set that is from a different dataset. **Test set** can be downloaded from:

https://www.kaggle.com/maciejgronczynski/biggest-genderface-recognition-dataset

The test set contains 27167 RGB images of varying size.

- You should resize your images to 100×100 while loading the images.



(a) Training and validation accuracy

(b) Training and validation loss

Figure 2: Examples of accuracy and loss plots

- You should have the same labels as your training and validation sets. In training and validation sets, the label names are 'Female' and 'Male', so modify the test set folders accordingly.

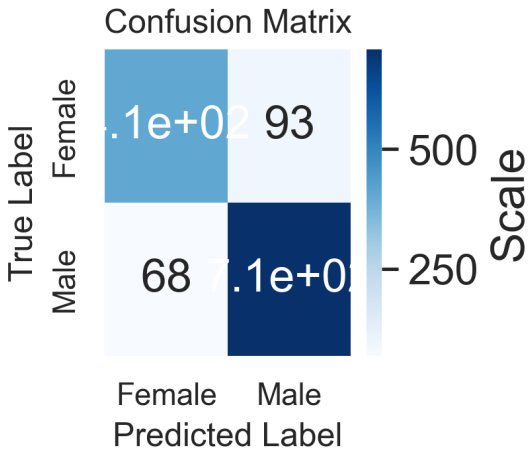
3 Important Dates and Deliverables

3.1 Report

A 4 page (excluding references, title page) group report should be submitted by **Monday October 4, 2021**. The report should include the following information:

- Title
- Group Number
- Student names and Student numbers
- A summary of your models (excluding the baseline model) - similar to the one provided in Fig. 3.
- A brief description of your experiments, including possible pre-processing steps, training, hyperparameters, activation functions, optimizattion/regularization techniques...or any other changes you made.
- The graphs/results requested in the task description, see Sec. 2.
- A discussion of the performance of your solution and how it relates to the literature used.
- All the .py code (that produces the results presented in your report).
- A paragraph at the end of your report outlining the ethical concerns that arise from this task as well as its limitations.

(a) Training and validation accuracy



(b) Training and validation loss

Figure 3: Examples of ROC curve and confusion matrix

3.2 Code

Your code should be a **plain Python script** which can be run to generate your predictions. You do not need to include the training data or the trained model. We strongly recommend you to use Keras.

3.3 Submission format

Put the **report and the code in a single zip file** named with your group ID, **e.g. group_1.zip** and submit it to the Canvas assignment page.

References

[1] Xu, Ziyi, Li Lu, and Pengfei Shi. "A hybrid approach to gender classification from face images." 2008 19th International Conference on Pattern Recognition. IEEE, 2008.

[2] Yang, Zhiguang, Ming Li, and Haizhou Ai. "An experimental study on automatic face gender classification." 18th International Conference on Pattern Recognition (ICPR'06). Vol. 3. IEEE, 2006.

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

[4] Yaman, Dogucan, Fevziye Irem Eyiokur, and Hazim Kemal Ekenel. "Multimodal age and gender classification using ear and profile face images." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. 2019.