# 01-Data

|  |  |
| --- | --- |
| **Module** | **Description** |
| 1\_images-build-v6.ipynb | Notebook to generate an array of machine font labeled images. |
| 1\_images-build-v6-test.ipynb | Notebook to generate a smaller size array of machine font labeled images. It is used to train the individual character classifiers. |
| 2\_train-gan.ipynb | Notebook to train GANs. |
| 3\_retrain-gan.ipynb | Notebook to continue GAN training. |
| 4\_generate-GAN-images\_array.ipynb | Notebook to generate an array of GAN images. The amount of images and the characters are defined by the user. |
| 4\_generate-GAN-images\_array-test.ipynb | Notebook to generate a smaller size array of GAN images. It is used to train the individual character classifiers. |
| 4b\_generate-GAN-images\_png.ipynb | Notebook to generate local stored images. It is used for checking image quality. |
| models | Folder with all the trained GAN models |
| characters | Folder with the results of GANs. There the quality of the training can be checked. |

# 02-Classifiers

|  |  |
| --- | --- |
| **Module** | **Description** |
| 1\_cv2-gan-build-images.ipynb | Notebook to create an array of machine font labeled images and GAN generated images. The amount of Gan images is user defined. |
| 1\_cv2-gan-build-images-test.ipynb | Notebook to generate a smaller size array of machine font labeled images and GAN generated images. It is used to train the individual character classifiers. |
| 2\_cv2-char-classification\_train.ipynb | Notebook to train 1 CNN for all characters by using only machine font data. |
| 2\_cv2-gan-char-classification\_train.ipynb | Notebook to train 1 CNN for all characters by using machine font data and Gan data. |
| 2\_cv2-char-classification\_train\_char.ipynb | Notebook to train 1 CNN per character by using only machine font data. |
| 2\_cv2-gan-char-classification\_train-char.ipynb | Notebook to train 1 CNN per character by using machine font data and Gan data. |
| 2\_cv2-char-classification\_train\_mixed.ipynb | Notebook to train specialized CNN using only machine labeled font data. Specialized means, models like capital vs small or underscore vs dash. The model train run for the specific notebook. |
| 2\_cv2-gan-char-classification\_train\_mixed.ipynb | Notebook to train specialized CNN using machine labeled font data and GAN data. The model train run for the specific notebook. |
| models\_char | Folder with CNN models per character trained only with machine font data and 32pixel size images. |
| models\_char\_gan | Folder with CNN models per character trained with machine font data and GAN at 32pixel size images. |
| models\_mixed | Mixed models trained only with machine font data. |
| models\_mixed\_gan | Mixed models trained with machine font data and GAN. |

# 05- Orchestrator

|  |  |
| --- | --- |
| **Module** | **Description** |
| 01-Data /Train characters | Train the GAN to generate font data   1. Epochs= training cycles 2. Pix= image pixel size in format (pix, pix), for small pix the 1\_images-build-v6.ipynb and 1\_images-build-v6-test.ipynb requires adjustment because the letters in the images are damaged. 3. characters\_all= an array with the characters that will be trained. There is one GAN per character. |
| 01-Data /Re-Train characters | Retrain the GAN (continuous learning)   1. Same parameters. |
| 01-Data /Generate fake images as array | Generate fake images as an array. Size is defined by the user.   1. number= amount of images to generate by character. 2. character= an array with the characters in images that will be generated. |
| 01-Data /Generate fake images of one character as png | Generate fake images stored locally for checking quality purposes.   1. pix= The size of the generated images in pixels. (pix, pix) format. 2. Number= the amount of images by requested character. 3. character= an array with the characters in images that will be generated. |
| 02-Classifiers/char classification (1 model for all characters) | Train 1 CNN for all characters by using machine font data or machine data and Gan data.   1. character = list of all characters 2. epochs\_class= the training CNN epochs 3. number= number of GAN images included in the dataset 4. pix= image size to train the CNN. |
| 02-Classifiers/char classification GAN and simple (1 model per characters) | Train 1 CNN per character by using machine font data or machine data Gan data.   1. character = list of all characters 2. epochs\_class= the training CNN epochs 3. number= number of GAN images for the target class (target letter) included in the dataset 4. numbertest= number of GAN images for the zero class (not the target letter) included in the dataset. 5. characters= array with target characters to train the CNNs 6. charact= array with the rest characters to train the CNNs |

# 04-Recognition

|  |  |
| --- | --- |
| **Module** | **Description** |
| recogntition.ipynb | Notebook to score the models and classify the image. |
| !!!! This part of the app is not ready. Use this Notebook as a reference of how to load, combine and score the CNNs to classify a character image. | |

# 03-Object Detection-GAN

|  |  |
| --- | --- |
| **Module** | **Description** |
| object\_detection.ipynb | Main notebook to run the app. |
| config.py | Path to tesseract. |
| functions\_char\_detection.py | Functions to preprocess word segments for getting the character segments. The function to cut the characters is inside the object\_detection.ipynb. |
| functions\_line\_detection.py | Functions to detect text lines in object segments. |
| functions\_object\_detection.py | Functions to detect text objects in images. For example paragraphs. |
| functions\_word\_detection.py | Functions to detect text words in line segments. |
| object\_output.npy | Notebook to generate local stored images. It is used for checking image quality. |
| data | Input data images. |
| app\_output | Application output. |

# 13-Object Detection-Tesseract

|  |  |
| --- | --- |
| **Module** | **Description** |
| object\_detection.ipynb | Main notebook to run the app. |
| config.py | Path to tesseract. |
| functions\_line\_detection.py | Functions to detect text lines in object segments. |
| functions\_object\_detection.py | Functions to detect text objects in images. For example paragraphs. |
| functions\_word\_detection.py | Functions to detect text words in line segments. |
| object\_output.npy | Notebook to generate local stored images. It is used for checking image quality. |
| data | Input data images. |
| app\_output | Application output. |